Using and Collecting Annotated Behavioral Trace Data for Designing and Developing Context-Aware Application

by

Yung-Ju Chang

**A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Information)
in the University of Michigan
2016**

Doctoral Committee:

Associate Professor Mark W. Newman, Chair
Professor Mark S. Ackerman
Associate Professor Natalie Colabianchi
Assistant Professor Predrag Klasnja

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my research advisor Mark W. Newman at the Interaction Ecologies Group for his great support and advice for my research work and career plan throughout my time as a Ph.D. student. Thanks for all the help, encouragement, critiques, feedback, advice, and time. Especially thanks to his guiding me back to my research when I was attempting to launch a startup in my third year of Ph.D. I also want to thank him for allowing me to explore different research topics over these years, including information seeking and environmental cognition. Although I am not able to include the pieces of work in this dissertation, my literature background in these areas are important knowledge assets allowing me to explore new directions at the intersection between these areas and my dissertation.

I want to thank all of the dissertation committee members: Mark Ackerman, Pedja Klasnja, and Natalie Colabianchi. It has been a wonderful intellectual experience working with them. I want to thank their thoughtful comments that improve the focus and the scope of the dissertation and suggest areas to explore in my future research. I want to thank John Tang, my mentor at Microsoft Research. I first want to thank him choosing me as an intern at Microsoft Research in 2012, for which I was able to start my journey in mobile communication and interruptibility, a very important research direction in this dissertation and of my future research agenda. It was an excellent experience working with John, and this internship was my first time starting to use a mixed-method approach to study mobile users' behaviors. Without his mentorship, I would not have been able to use these methods in my other research studies. I also want to thank Soo Young Rieh and Malcolm McCullough, my committee members of the preliminary

exam, for providing me with thoughtful comments and feedback on the topic of information seeking in the urban environment from different perspectives.

I want to thank all of the (former) colleagues at the Interaction Ecologies Group, who collaborated with me and contributed to the projects I managed. In no particular order (probably omitting some): Dong Tao, Rayoung Young, Pei-Yao (Perry) Hung, I-Chun Hsiao, Manchul Han, Chuan-Che (Jeff) Huang, Shriti Raj. Special thanks to Rayoung, who has been with me, been thoughtful and considerate, and encourages me when I was stressed throughout this journey. I have been enjoying the time discussing our own research with Rayoung. It is my best pleasure to graduate with you at the same time! I want to thank all of friends and colleagues who helped the Ubicomp labeling study, including Hsin-Ying Wu, Hsin-Yu Lin, and Noureen Dharani. Thanks to all faculty members who have been teaching me skills and knowledge in the past few years. Thanks to colleagues in the ACM SIGCHI community for their inputs and discussions on my research projects at conferences. Thanks to the research and administration staff at UMSI for providing reliable assistance for my research. Thanks to Judy Dyer, Christine Feak, and Pamela Bogart who helped me improve my English skills and this dissertation through their thoughtful and thorough feedback. Thanks to Shih-Hsuan Chou, Min-Chih Liu, Surong Ruan, and Lezong Li, who started the language learning project with me, and Yi-Wei Chia, Kevin Chang, Kerry Kao, and Morgan Chen who were willing to continue this project, although we could not make it eventually. It was a pity that we were not able to continue the project together, for now, because my choice of focusing on my dissertation. But it has been a great experience in working with you guys. And I believe at some point the plan is going to work out.

Last but not least, thanks to my dear family members and friends. Without your love, support, and encouragement, this wouldn't have been possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Ubiquitous Computing has been a focus of numerous researchers hoping to create environments where users are served by heterogeneous computing devices responding to their contexts. Thanks to these researchers' research efforts, computing infrastructures, sensing devices, and intelligent systems have been developed, making the creation of context-aware systems more viable, economic, and appealing to designers and developers. This thesis aims to respond to this emerging trend by developing systems and practices supporting more effective development of context-aware applications. In particular, I focus on using a *capture-and-playback* approach—capturing and playing back behavioral and contextual data to prototype, test, and evaluate context- aware applications. The thesis makes five main contributions in this area. The first two contributions focus on supporting playback. In Chapter 3, I present findings and lessons learned from two case studies and a developer study involving the capture-and-playback approach and tool, of which the results inform the design space for supporting context-aware application development. Second, I present a design, development, and evaluation of a capture-and-playback toolset called CaPla, which support different activities in developing context-aware applications.

Starting from Chapter 4, 5, and 6. I present my research efforts making three contributions to data capture. First, I present findings from an empirical study investigating smartphone users' mobile receptivity to incoming communications. The findings indicate factors to be considered when sending data capture requests to smartphone users. In Chapter 5, I present a field study investigating the effectiveness of using three different approaches for collecting personal behavioral and contextual data. The results show pros and cons of the three approaches, as well as smartphone users' behaviors in using the approaches and

how activity impacts users' data collection behaviors. Finally, in Chapter 6, I present a configurable, flexible, and extensible mobile data collection tool called Minuku. Minuku can monitor complex contextual conditions, schedule and perform highly situated actions, and allows performing different styles of data collection approaches.

The findings of the studies and the experiences with the systems point towards the design space for a more comprehensive capture-and-playback tool and a set of practices of performing a capture-and-playback approach.

Chapter 1. Introduction

Mark Weiser introduced the area of ubiquitous computing and envisioned an environment where users are surrounded and served by augmented and heterogeneous computing devices and computational resources (Weiser, 1991). One illustration in this vision is people's constant access to information, offered by the surrounding and widespread computational resources. Another illustration is the emergence of new applications that introduce a new paradigm of interactions: natural interfaces between humans and computations that offer a variety of communications; automating capture of and universal access to people's live experiences; and context-aware applications that adapt their behavior based on the information sensed from physical and computational environments (Abowd & Mynatt, 2000). Context-aware applications, such as a mobile application providing information of local services based on the users' current location and mobility information (e.g. Google Now[1]), or a thermostat adjusting temperature based on the occupancy status of households at home (e.g. Nest[2]), moves human-computer-interaction from desktop into a variety of environments like living rooms, health clinics, museums, vehicles, and streets.

As the venues for human-computer interaction diversify, designers and developers of context-aware applications face new challenges in understanding users' existing practices and needs, and in designing appropriate systems. Today, designers not only need to understand the contexts of use but also need to devise

---

[1] https://www.google.com/landing/now/

[2] https://nest.com/

the ways that the application will adapt and respond to changing contextual conditions. Despite the increased demands, it remains a great challenge to recreate the anticipated context of use during development time. During the development of context-aware applications, it can be difficult to prototype, test, and evaluate the application's behavior before it is deployed into the field.

To address this challenge, a number of systems have been developed to support the development of context-aware systems in a variety of ways: through support for rapid prototyping of pervasive computing (Carter, Mankoff, & Heer, 2007; Hartmann, Abdulla, Mittal, & Klemmer, 2007; Li, Hong, & Landay, 2004), field testing lo-fi prototypes (Hartmann et al., 2007; Li et al., 2004) and Wizard-of-Oz support for early stage tests involving context awareness (Carter et al., 2007; Hartmann et al., 2006; Li et al., 2004; MacIntyre, Gandy, Dow, & Bolter, 2004). These works, however, focus more on "bringing the lab into the field"— bringing prototyping, testing, or evaluation into the field—and do not address the issue of considerable cost entailed in iteratively conducting these development activities in the field. When developing applications, the bulk of the work involved in designing, developing, and evaluating applications does not happen in "the field," but rather in offices, cubicles, studios, and labs, where designers and developers are involved in the reflective activities of design-build-test. To allow designers and developers to more easily recreate contextual conditions their systems have to encounter at runtime during design and development while also reducing the cost for development, tools that help "bring the field into the lab" are needed.

As a step in this direction, RePlay (Newman et al., 2010) was developed to bring captured user behavioral traces into the prototyping process to help with rapid design, development, and testing of new functionality under realistic contextual conditions. Specifically, RePlay supports testing context-aware applications by its capture-and-playback (C&P) feature—capturing contextual data in the field and

playing the collected data back to an application under development in the lab. This feature allows an application to be tested and evaluated under different contextual conditions. The C&P feature, however, has not been formally evaluated as to how it does in fact support different design and development activities of context-aware applications, such as prototyping, testing, and evaluation. It also remains unclear what challenges designers and developers would encounter when using a C&P approach and a tool like RePlay to design and develop context-aware applications.

The first goal of this thesis, therefore, is to evaluate using a C&P approach and tool (in this case, RePlay) to design and develop context-aware applications, and explore what other features a C&P tool should equip to better support the design and development activities. Through this investigation, I aim to inform the design space for a C&P tool to better support the design and the development of context-aware applications. I address this goal in Chapter 3.

The second goal of this thesis is motivated by two key findings derived from the investigation. The first finding is identifying a crucial role of annotations on the captured data—the metadata describing the characteristics of the data—in the design process. Annotations not only help a design team more effectively use captured data throughout the entire design process, but also facilitate communication among the design team members. The second finding is the value of crowdsourcing data capture to the mobile crowd—a crowd of mobile phone users who have sensing devices to record their behavioral and contextual traces. Crowdsourcing data capture to the mobile crowd can potentially increase the amount of data for use during development; ease the burden of data capture by developers and designers themselves; and enhance the variety and diversity of data. In particular, it important to conduct such a data collection in real life settings, so that the data collected can well represent realistic behaviors and

contextual conditions. This is especially important for data that will be used for developing context-aware applications. However, collecting individual behavioral and contextual data via the mobile crowd in real-life settings has been underexplored. Today, it remains unclear what would be a good practice of crowdsourcing collecting individual behavioral and contextual data to the mobile crowd and what features a data capture tool should equip to better support such a data collection.

The second goal of this thesis, thus, is to contribute to the research of collecting individual behavioral and contextual data with annotations, including: (1) exploring a good practice of collecting annotated behavioral and contextual data; (2) exploring the design space for a capture tool to more effectively capture behavioral and contextual data; (3) exploring what a good time would be to send a data collection request to mobile users by understanding their interruptibility and receptivity on smartphones in relation to incoming communication requests; and (4) building a configurable, flexible, and extensible mobile data collection tool to support collection of behavioral and contextual data,

In this thesis, I address the subgoals (1) and (2) with a field study described in Chapter 5 that investigates the use of three different approaches for collecting annotated travel activity data. The field study along with an empirical study described in Chapter 4 address the subgoal (3). Finally, I address subgoal (4) by an implementation of an Android mobile collection tool called Minuku described in Chapter 6.

Having addressed these goals, the conclusion of this thesis is presented as a thesis statement and a set of contributions in the next section.

1.1 Thesis Statement and Contributions

The thesis statement is:

An effective and efficient use of captured behavioral and contextual trace data for designing and developing context-aware applications is achievable through a combination of

1. *a capture-and-playback system facilitating prototyping, testing, and evaluation with the support of visualizing, filtering, selecting and modifying behavioral trace data*

2. *a set of good practices designers and developers can follow to effectively request the mobile crowd to collect annotated behavioral and contextual traces they need.*

3. *a tool that can be customized with flexibility and extensibility to collect behavioral and contextual data for various needs.*

Contributions of the thesis are:

a) Findings and lessons learned for informing the design space for supporting context-aware system development – This is developed through reflections on two case studies of context-aware systems, and a developer study of a capture-and-playback tool.

b) A capture-and-playback tool called CaPla – CaPla addresses three themes important to context-aware system development—selecting examples, manipulating data, and iterative testing—through support for visualizing, filtering, selecting and modifying behavioral trace data. An evaluation is conducted to examine effectiveness of these themes.

c) Improved understanding of mobile interruptibility and receptivity – The empirical study in Chapter 4 provides insights into mobile users' interruption management practices on smartphones, and characterizes how such practices affect their attentiveness and responsiveness to incoming communication.

d) Analysis of approaches to collecting annotated activity data through the mobile crowd – The field study in Chapter 5 suggests the pros and cons of using three different approaches to collect annotated travel activity data; uncovers the impact of activity on users' collection behavior as well as their receptivity to annotation tasks in the field; and provides design and methodological suggestions on the approach and tool for collecting annotated activity through the mobile crowd.

e) A mobile data collection tool called Minuku – Minuku described in Chapter 6 provides a framework that makes it configurable, flexible, and extensible. It is capable of monitoring complex contextual conditions; scheduling and performing highly situated actions, and performing different approaches to collect annotated activity data.

Figure 1.1. shows three research areas this thesis makes contributions to, and the position of each of these five contributions relative to the areas. Overall, (a) and (b) contribute to the design space for supporting the development of context-aware systems; (c) mainly contributes to mobile interruptibility and receptivity; (d) and (e) mainly contribute to mobile crowdsourcing aimed at supporting the development of context-aware systems, especially where these intersect with mobile interruptibility and receptivity.

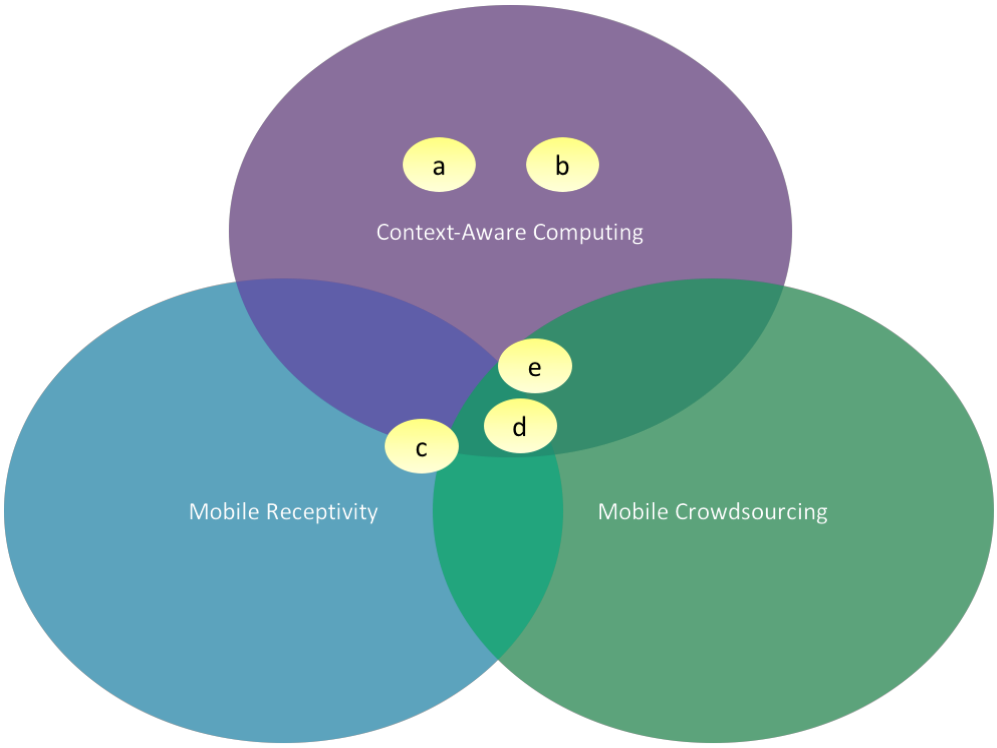


Figure 1.1 The research areas and the relative position of this thesis' contributions.

1.2 Dissertation Outline

This thesis is structured as follows. In ***Chapter 2***, I survey the relevant background literature on using captured behavioral and contextual traces in context-aware application design; supporting data collection in mobile crowdsensing/sourcing; mobile sensing systems; collecting annotations on behavioral and contextual data; and mobile interruptibility, receptivity, and opportune moments. In ***Chapter 3***, I present findings and lessons learned from two case studies and a developer study of the C&P tool RePlay, as well as a

resulting capture-and-playback tool called CaPla of which the features are informed by the findings. In ***Chapter 4***, I describe the study investigating mobile users' interruption management practices and their attentiveness and responsiveness to incoming communication request on mobile phones. In ***Chapter 5***, I present the field study investigating the use of three different approaches for collecting travel activity data with annotation. In ***Chapter 6*** I describe the Minuku system. In ***Chapter 7***, I conclude the thesis with discussions of future directions for supporting context-aware application development.

Chapter 2 Literature Background

2.1 Using Captured Behavioral and Contextual Data in Interaction Design

The importance of collecting data about users to inform system design is well understood by interaction designers and, indeed, forms one of the central tenets of user-centered design (Rogers, Sharp, & Preece, 2011). Design ethnography (Salvador, Bell, & Anderson, 1999) and contextual inquiry (Beyer & Holtzblatt, 1997) are commonly practiced techniques that aim to collect and synthesize rich data about users' behaviors and contexts for guiding design. The sort of direct observation required by ethnographic methods is extremely effective for understanding users, but can present challenges in terms of the effort required, the situations that can be effectively studied, and the kinds of data that can be collected.

Collecting data about users and their contexts is a standard starting point for user-centered design practice, wherein such data is used to produce requirements documents, personas, and scenarios that subsequently influence later design stages. A number of systems have sought to enrich the set of tools for data collection and analysis, focusing on the use of behavioral and contextual data such as sensors and *in situ* self-reports to generate a comprehensive picture of user behaviors and contextual conditions. For example, Digital Replay System (DRS) (Greenhalgh, French, Tennent, Humble, & Crabtree, 2007) and ChronoViz (Adam Fouse, Weibel, Hutchins, & Hollan, 2011) both provide tools for capturing and visualizing user behaviors to support analysts in making sense of user behaviors. MyExperience (Froehlich, Chen, Consolvo, Harrison, & Landay, 2007a) and Momento (Carter et al., 2007) support rich data capture, and go beyond DRS and ChronoViz in terms of system design support by, for example, enabling context-triggered experience sampling (ESM) (Csikszentmihalyi &

Larson, 1987) and creating low-fi prototypes, and by enabling remote orchestration of field-based user tests. However, the goals of these systems have been helping analysts and designers gain a better understanding of user behaviors and the contexts in which the system is used. Despite these valuable features and the fact that the data collected in these systems, in principle, could be repurposed for system development, the aforementioned systems have not been tested and evaluated in supporting the development of context-aware applications.

Researchers have developed a set of systems for supporting the development of context-aware applications. Some of the systems focused on developing simulators that simulate contextual conditions not based on collected data (Barton & Vijayaraghavan 2003; O'Neill et al., 2005). However, these simulators have not gained much attention, probably because the phenomena that must be simulated are difficult to model. Several other systems have looked at helping developers use collected data in the course of implementing machine learning algorithms—an essential activity in creating a context recognizer. For example, A CAPpella (Dey, Hamid, Beckmann, Li, & Hsu, 2004) is a programming-by-demonstration system for specifying application events that should be recognized by the system when particular sensed conditions occur. Exemplar (Hartmann et al., 2007) allows designers to train gesture recognizers by demonstrating user actions. Gestalt (K. Patel et al., 2010) provides a general-purpose support for implementing and testing machine learning algorithms, helping programmers understand and debug the interactions between their code and the data. While these tools have utilized collected data for training a system to recognize the demonstrated behaviors, they require implementation of a machine learning based learning system and require controlled lab-based data collection. These requirements differ considerably from systems supporting lighter weight development activities (e.g., heuristically-based context interpretation) and rapid

exploration of potential functionality, and supporting projects involving opportunistic data collection.

Another set of tools focus on using collected data in reflective prototyping (Hartmann et al., 2006), a rapid iterative loop of "design-test-analyze," rather than focusing on support for formal testing and verification. These systems recognize the value of recreating contextual conditions using the data collected in situations resembling the anticipated context of use and playing back the data to enable rapid iteration of sensor-driven interactions. Exemplar (Hartmann et al., 2007) is one of the examples mentioned earlier. Other tools that support prototyping with captured data and playback include: DART (MacIntyre et al., 2004), Activity Designer (Li & Landay, 2008), Panoramic (Welbourne, Balazinska, Borriello, & Fogarty, 2010), and RePlay (Newman et al., 2010). Specifically, DART allows data collected during user tests to be incorporated into subsequent prototyping activities to enable designers to design applications. Panoramic supports the use of captured data for verification of end-user created rules that define complex events. ActivityDesigner supports the playback of sensor traces captured both before and during prototyping as part of applications development involving activity recognition.

While these tools provide features to support capture-and-playback as a part of full-featured prototyping tools, RePlay (Newman et al., 2010) is the first tool focusing on capture-and-playback as a first order concern and offers support for capturing, organizing, transforming, and playing back behavioral and contextual data throughout the design and development lifecycle, independent of the choice of prototyping or development tools. RePlay, however, has never been formally evaluated in terms of how its features do support prototyping, testing, and evaluating context-aware applications. It is also unclear what challenges designers and developers would face in using a capture-and-playback approach and a tool in

developing context-aware applications. It is the goal of Chapter 3 to answer these research questions.

2.2 Supporting Data Collection in *Mobile Crowdsensing/sourcing*

Mobile sensing, generally speaking, refers to collecting data through mobile sensing devices such as smartphones. Two main classes of mobile sensing discussed in the literature are: *personal sensing* and *mobile crowdsensing* (MCS). Although the literature on personal sensing often also involves a number of participants, which can also be considered leveraging a *crowd* of people, the focus of the literature of this class is mainly on capturing information of, or related to individuals via the sensing systems, the purpose including understanding individual behavior (Mulder, Ter Hofte, & Kort, 2005; S. N. Patel, Kientz, Hayes, Bhat, & Abowd, 2006), understanding a phenomenon (e.g. Min, Wiese, Hong, & Zimmerman, 2013), evaluating a system for monitoring personal informatics (e.g. Dickerson, Gorlin, & Stankovic, 2011; Mun et al., 2009), and promoting personal wellness (e.g. Lane, Mohammod, et al., 2011).

In contrast, the literature of mobile crowdsensing focuses on using mobile systems to sense and record information relating to public phenomena. As the result, the purpose of recruiting a crowd of participants is for dividing a large and a time-consuming data collection task to a number of people, whose collected data when are aggregated can display some nature of a public phenomenon, information, and environment of interest. In literature, the main categories of data being collected include but are not limited to: air quality condition (K. Hu, Wang, Rahman, & Sivaraman, 2014; Kanjo, Bacon, Roberts, & Landshoff, 2009; Paulos, Honicky, & Goodman, 2007); road and traffic condition (Ilarri, Wolfson, & Delot, 2014; Mohan, Padmanabhan, & Ramjee, 2008; Thiagarajan et al., 2009; X. Zhang, Gong, Xu, Tang, & Liu, 2012; Zhu, Li, Zhu, Li, & Zhang, 2013); noise and sound level (D'Hondt, Stevens, & Jacobs, 2013; Kanjo, 2010; Lu, Pan, Lane,

Choudhury, & Campbell, 2009; Maisonneuve, Stevens, Niessen, Hanappe, & Steels, 2009; Rana, Chou, Kanhere, Bulusu, & Hu, 2010; Stevens & D'Hondt, 2010); price of products (Deng & Cox, 2009; Dong, Kanhere, Chou, & Bulusu, 2008); parking information (Coric & Gruteser, 2013; Villanueva, Villa, Santofimia, Barba, & Lopez, 2015); and transit tracking (Farkas, Feher, Benczur, & Sidlo, 2015; Thiagarajan, Biagioni, Gerlich, & Eriksson, 2010; Tomasic, Zimmerman, Steinfeld, & Huang, 2014; Zhou, Zheng, & Li, 2012; Zimmerman et al., 2011a).

In mobile crowdsesning, one area of work particularly relevant to this thesis is systems supporting participatory sensing (Kanhere, 2011). Participatory sensing was introduced by the CENS group (Burke et al., 2006) where the context was citizen scientists using mobile sensing technology for environmental monitoring. In participatory sensing, users who are requested to collect data initiate data collection with a guideline provided by the task requester. Participatory sensing is considered useful when the data to be collected include subjective and qualitative information, comments, and annotations, which cannot be obtained from physical sensors (Sakamura, Yonezawa, Nakazawa, Takashio, & Tokuda, 2014). Using a similar concept, *citizen science* is defined as a way to harness the power of the public, or a form of research collaboration between researchers and volunteers, to solve real-world problems or answer scientific questions (Cooper, Dickinson, Phillips, & Bonney, 2007; Silvertown, 2009). Despite the different terms being used, participatory sensing and citizen science share a common ground that both involve and rely on a number of people to actively participate in contributing data. In contrast to these two concepts, opportunistic sensing refers to the instrument passively collecting data in the background, usually without users' participation and awareness (Khan, Xiang, Aalsalem, & Arshad, 2013; Khan et al., 2013; Lane et al., 2010). Due to this difference, while systems aimed at supporting participatory sensing or citizen science are not essentially different from each

other in terms of systems feature, systems aimed at supporting opportunistic sensing usually do not consider supporting user interaction in the system design.

Additionally, *mobile crowdsourcing* is another emerging stream of research referring to crowdsouring tasks to mobile users via a mobile system (Konomi & Sasao, 2015), or a system in the field (Goncalves, Hosio, Ferreira, & Kostakos, 2014; Heimerl, Gawalt, Chen, Parikh, & Hartmann, 2012; Hosio, Goncalves, Lehdonvirta, Ferreira, & Kostakos, 2014). The idea of mobile crowdsourcing is to overcome the limitation of online crowdsourcing in performing tasks beyond the desktop. As a result, although tasks in this stream of research also include those typical in the online crowdsourcing projects such as annotating images and videos (e.g. Hosio et al., 2014), it is featured for crowdsourcing tasks only achievable in the field such as reporting local events (Agapie, Teevan, & Monroy-Hernández, 2015), which are not essentially different from collecting public information as in mobile crowdsensing projects. However, it should be noted that collecting annotated "personal" behavioral and contextual trace data, one of the foci in this thesis, is beyond "sensing public phenomenon or information." Personal behavioral data are more difficult to validate and assess than public data. On the other hand, despite the fact that mobile crowdsourcing accommodates more types of task, it is also not as specific as mobile crowdsensing in terms of conveying the type of task being performed (i.e. sensing). In this thesis, I tend to say "leverage the mobile crowd to collect behavioral and contextual data" or "crowdsourcing data collection to the mobile crowd" instead of using either term.

However, since literature in both streams of work is relevant to this thesis, the literature review will include both mobile crowdsensing and the mobile crowdsouring projects of which the tasks are related to collecting behavioral and contextual data. In particular, I will concentrate on the literature related to *supporting data collection* in both streams. The literature includes supporting

participants and improving data quality, the latter including participant selection and task distribution to the appropriate mobile crowd. Because of the focus on supporting data collection, I will not review the literature in incentive design for mobile crowdensing, I will also not give a comprehensive review of the types of crowdsensing applications because most of them are not related to supporting data collection, but instead, related to evaluating a specific system in a specific sensing domain. Several surveys of incentive mechanism for mobile crowdsensing are available in Arakawa & Matsuda, (2016); Gao et al., (2015); Jaimes, Vergara-Laurens, & Raij, (2015), and Restuccia, Das, & Payton, (2015). Surveys of types of mobile crowdsensing applications are available in (Ganti, Ye, & Lei, 2011; Khan et al., 2013; Lane et al., 2010), Wang et al. (2015), and (Thebault-Spieker, 2012).

2.2.1 Support Participants

Several systems for supporting participatory sensing have been developed. Because the primary role in participatory sensing is data collectors, systems aimed to support participatory sensing mainly focus on supporting data collectors.

2.2.1.1 Energy Saving

Energy consumption on the phone is a critical factor affecting mobile users' willingness to participate in crowdsensing tasks. Research has shown that people are more willing to participate in crowdsensing tasks if the tasks have limited impact on the battery lifetime of their phones (Foremski, Gorawski, Grochla, & Polys, 2015). To reduce the negative impact of battery consumption on mobile users' participation, a large body of research in mobile sensing proposes different strategies to reduce the power consumption of mobile sensing tasks. One of the most common solutions is deactivating, or lowering the sampling of sensors when the sensors are not in used or when obtaining high granularity of sensor data is not

necessary. For example, Jigsaw (Lu et al., 2010) is a framework that uses a pipeline architecture to manage sensors. It activates and deactivates sensors depending on the need and resource availability. It also adjusts sensor sampling for reducing power consumption. Roy et al. (Roy, Misra, Julien, Das, & Biswas, 2011) used a sophisticated model to capture the tradeoff between an estimated accuracy of sensor data and the overhead incurred in acquiring the necessary sensor data. In computing the tradeoff results, they aimed to obtain the best set of sensors to contribute to context determination.

Another stream of research in this area focuses on adapting the sampling rate of GPS locations. GPS is known as an energy-expensive source. Continuous request for location updates from GPS is likely to deplete the battery of users' phones within several hours. As a result, a number of works have sought to reduce the power consumption of GPS by obtaining location data from alternatives such as location sources such as a Wifi or a cellular network. However, locations obtained through these sources are generally less accurate then location obtained from GPS[3]. As a result, when using these sources, researchers may obtain less reliable and accurate location data about the users. To address this issue, researchers have proposed switching location sources between GPS and a cellular network (Paek, Kim, Singh, & Govindan, 2011; Zhuang, Kim, & Singh, 2010). Another common approach is enabling or only requesting GPS location data when necessary, such as turning on the GPS only when the accuracy of GPS is higher than a Wifi or a cellular network (Paek, Kim, & Govindan, 2010), or when the users are detected to be moving based on accelerometers (Paek et al., 2011; Zhuang et al., 2010) or based on cellular signal changes (Foremski et al., 2015)

[3] http://developer.android.com/guide/topics/location/strategies.html

Adapting data uploading frequency to save power has also been adopted. For example, Musolesi et al (Musolesi, Piraccini, Fodor, Corradi, & Campbell, 2010) evaluated several techniques to optimize a data uploading process. The techniques included calculating the tradeoff between transmission overhead and accuracy, as well as using a location-based uploading strategy. They showed that these techniques could save battery life of mobile phones and improved the performance of continuous mobile sensing.

In addition, one known source of power consumption of mobile sensing is waking up the phone from an idle state for collecting data. To reduce the power consumed by this process, Lane et al., (2013) proposed collecting data only when the phone has been woken up by the user, such as the user using an application, the user being on a call, or the phone having already performed other sensing tasks. They collected data from 1,320 smartphone users and show that this method effectively collected mobile sensor data from these users while using up to 90% less energy. However, one limitation of this approach is the uncertainty of when and for how long sensor data would be collected. It would not work well in cases where researchers want to capture a complete trace of users.

Finally, unlike the tools mentioned above aimed at saving energy, SystemSens (Falaki, Mahajan, & Estrin, 2011) is a system aimed to capture resource usage such as CPU, memory, and battery in smartphone research deployments to inform researchers about the energy consumption of a system. In addition, SystemSens also saves power by uploading data only when the phone is charged.

Although the goals of these works are not directly related to the research questions of this paper, they provide useful strategies for increasing the battery life of users' mobile phones. These works thus inspire Minuku, a research tool

described in Chapter 6, to adjust the frequency of location updates by the activity detection method.

2.2.1.2 Privacy

Protecting the privacy of participants is also important to mobile crowdsensing projects aiming at collecting behavioral and contextual data from mobile users. A number of works have sought to protect participants' privacy using pseudonyms, i.e. aliases instead of real names (Shilton, 2009; Shilton, Burke, Estrin, Hansen, & Srivastava, 2008). Christine et al. (Christin, Reinhardt, Kanhere, & Hollick, 2011) analyzed privacy protection in a number of participatory systems and argued that privacy threats do not only exist in personally identifiable information such as phone number and email address, but also in time and location, sound samples, pictures and videos, and even accelerometer readings, environmental data, and biometric data. Christine et al. suggested several steps to protect the privacy of participants. The first is to control the data collection process at the participant level, including allowing participants to express their privacy preferences such as to selectively enable certain sensors depending on their current location and current social surroundings, and allowing participants to adjust data collection frequency (e.g. from every five seconds to every thirty seconds) to decrease the granularity of data. However, researchers then face a tradeoff of collecting low-fidelity of data. Researchers can also choose to discard any data that participants do not indicate their willingness to upload (Shilton et al., 2008). However, this method requires participants to constantly indicate their willingness and can add an extra burden to them.

Second, it is important to anonymize task distribution to participants. Research in this direction includes: using task beacons to identify participants instead of requiring participants to register to a central server (Kapadia, Kotz, &

Triandopoulos, 2009); downloading the tasks in densely populated location to make it difficult to identify participants from high density of people (M. Shin et al., 2011); processing the authentication of participants by asking them to show their membership of a particular registered crowdsensing platform; and hiding participants' location using a certain routing method (M. Shin et al., 2011). Researchers such as Mun et al., (2009) also suggested providing an option for participants to hide sensitive location information during data reporting.

In data aggregation, Shi et al. proposed letting participants to mutually protect each other's privacy instead of relying on a central management unit. Specifically, participants transmit only partial of their data to the server and distribute the rest to other participants before transmitting to the server (Shi, Zhang, Liu, & Zhang, 2010). This distribution largely decreases the likelihood of attributing one piece of data to any single participant. In data processing, researchers can render data containing privacy-sensitive information such as photos or sound indistinguishable, so that the data does not directly reveal the identity of the participants.

Finally, while data submitted to a central server are likely to be accessed, shared, and utilized by a group of people needing the data, researchers have also proposed allowing participants to specify intended audience of specific set of data, either a group or individuals. (Gaonkar, Li, Choudhury, Cox, & Schmidt, 2008; Kansal, Nath, Liu, & Zhao, 2007; Shilton, 2009). Researchers also have proposed allowing participants to specify a specific criteria (e.g. time) under which their data are accessible (Shilton et al., 2008). Even after the data have been released, participants can monitor the access of the data, including when the data are accessed and who access the data (Shilton, 2009; Shilton et al., 2008).

The literature, in general, suggests a set pf good practices of preserving participants' privacy when collecting their behavioral and contextual data. Since a long-term goal of this thesis is to enable more effective data capture, preserving privacy is an important step to take to increase participants' willingness to continually contributing their behavioral and contextual data.

2.2.1.3 Other Supports

There are other supports mobile crowdsensing systems and platforms provide. For example, Sakamura et al. (2014) aimed to help participants choose sensing tasks worth to contribute by visualizing the importance of the tasks through quantifying them based on physical sensors and a sensor visualization on a map. Another type of support is facilitating communication between researchers and participants. For example, Pogo (Brouwers & Langendoen, 2012) is a middleware framework allowing direct interaction between researchers and participants. That is, instead of having a central server serving to manage data transfer data requesters and participants, Pogo's central server functions merely as a communication switchboard between data requesters and participants. Thus, data requesters can directly interact with participants on their devices without needing to log in a central server. @migo (Bachiller et al., 2015) is another framework providing communication support between data requesters and participants. The challenge the system aims to address is a barrier to reaching participants when participants move across devices. @migo addresses this challenge by using online social networks (OSNs) such as Facebook and Twitter at the middleware level to reach and communicate with participants, assuming that participants would install OSN software across mobile devices.

2.2.2 Assessing Participants' Data Contributions

Researchers have explored different ways to evaluate participants because the outcome can be useful for data requesters to select suited participants for specific data collection tasks. Reddy et al. (2008) is an early work in participatory sensing seeking to develop a set of metrics to determine participants' performances in sensing projects. The authors suggested that contribution should be defined according to a number of qualities, including the sensor type and the modalities being used in the task; the spatial and temporal context in which the task is performed; and timeliness, relevance, and quality of the collected sensor data. Specifically, timeliness indicates the latency between when a phenomenon is sampled and when it is available to a sensor data processing module. Relevance indicates how well the sensor data sample describes the phenomenon of interest. Quality includes the probability of detection, probability of a false positive, or probability of a false negative. They also proposed metrics for evaluating participants' responsiveness to the task request, the amount of tasks they take, the frequency they check in with the crowdsensing system, whether they upload data regularly, and whether they take privacy precautions with their data such has blurring third party images in photographs.

Other researchers sought to measure the amount of noise in collected sensor data. An accurate and reliable noise measurement can lead to reliable evaluation of participants' contribution. Xiang et al. (2013), for example, aimed to quantify sensor noises in collected pollution data using the confidence interval. Specifically, Xiang et al. used an EM (Expectation Maximization)-based iterative estimation algorithm to compute the maximum likelihood estimation (MLE) of sensor noise. Then they leveraged the asymptotic normality of MLE and the Fisher information to compute the confidence interval. Their results showed a success rate where the true values of sensor noise fall into the 95% confidence interval. Also using the EM algorithm, their subsequent study improved the noise

measurement and also helped calibrate sensor in monitoring pollution (Xiang, Yang, Tian, Cai, & Liu, 2015).

2.2.3 Participants Selection, Recruitment, and Task Distribution

Selecting well-suited participants among a large number of users and distributing tasks to them is an essential yet challenging process in mobile crowdsensing. Being well-suited means being able to collect data with high quality. We have shown earlier that data requesters have explored ways to evaluate participants. It is important, then, to recruit and select participants based on the evaluation results. The results can also potentially help data requesters classify and analyze the collected sensor data based on the reputation of the participants (Yang, Zhang, & Roe, 2011).

Numerous reputation frameworks have been proposed, most of which consider participant's data contribution. For example, Huang et al. (2010a) computed a device reputation score to reflect trustworthiness of participants' contributed data in the context of noise monitoring. Reddy et al. (2010)'s reputation calculation considered participants' willingness of collecting data (whether data collected when the opportunity is given) and their diligence in collecting data (timeliness, relevance, and quality of data). Truskinger et al.(2011) proposed a reputation framework using a combination of an initial score based on participants' self-assessment via a questionnaire and a performance score based on the time efficiency, accuracy of data, and validity of data. The authors argued that the consideration of initial score helped them predict participants' performance with 90% accuracy.

Other researchers have sought to compute participants' reputation while preserving their privacy. For example, X Wang et al (2013)'s reputation framework separated data reporting process from reputation calculation. The

framework uses a blind signature that hides participants' identity in each data report and disallows the server to associate multiple reports to same participants. Also using a blind signature, Christin et al. (2013) proposed periodically generating pseudonyms and then transferring reputation of participants among one another. Ren et al. (2015) considered participant's reputation in crowdsourcing, task delay, and their social attributes. The social attributes in their study were defined as "the characteristics or features of an individual in his social life," including participants' interests, friend circle, and the living area.

Researchers have also utilized an auction mechanism to help selection of participants. For example, Kantarci and Mouftah (2014) used an auction-based reputation system to allow participants to bid tasks. In their reputation framework, participants' reputation is a function of the accuracy of collected data. A reputation score is taken as an input to the system, which generates an output representing an overall utility of the selected participants and the average utility per participant. The research used a simulation to show that using such an auction mechanism improved the overall utility of the platform while degrading the ratio of maliciously crowdsourced tasks by 75%.

In addition to using a reputation framework to help select participants, the other main direction on participant selection is identifying suited participants (or mobile devices) according to a set of predefined criteria. In this direction, researchers have used various criteria, including mobile users' current location (Linnap & Rice, 2014), speed (Das, Mohan, Padmanabhan, Ramjee, & Sharma, 2010), mobility behavior such as transportation and moving traces (He, Cao, & Liu, 2015; Konomi & Sasao, 2015; Sasank Reddy, Shilton, et al., 2009), sensing capability of the device (Das et al., 2010; Sasank Reddy, Samanta, et al., 2009), and available resources such as battery lifetime, currently executed tasks, or processing capabilities (Sasank Reddy, Samanta, et al., 2009).

It is noteworthy that although a number of research works in this direction adopt similar criteria for selecting participants, their goals are not necessarily the same. For example, Reddy et al's works (2009, 2010), probably the earliest ones exploring the area of participant selection and recruitment in mobile crowdsensing, proposed a coverage-based selection strategy in order to maximize the spatial coverage of collected data. Also for maximizing the coverage of collected data, Cardone et al. (2013) proposed a mechanism with a predefined number of participants, and Singla and Krause (2013) imposed an additional constraint of total incentive on participant selection. In contrast, D. Zhang et al. (2014) proposed a participant selection framework of which the goal was to minimize the total incentive payments under a coverage constraint. To minimize the total incentive, they aimed at selecting a minimal number of participants. However, their method still ensured a predefined coverage in each sensing cycle. For a similar purpose of reducing the number of participating in a sensing task, Ahmed et al (2011) used a discrete Markov chain to model participants' mobility for selecting participants while ensuring that at least a certain percentage of the targeted area was covered in a certain time. Hachem et al. (2013) also modeled participants' mobility to determine whether to register a specific participant's device based on the probability of other devices being present at the locations of their expected path.

Furthermore, to address the issue that participants (or devices) assigned a task are not necessarily available for, or capable of performing the task (e.g. the mobile phone is not equipped with the sensors required by the tasks), researchers have proposed ways to transfer the task to other participants, such as transferring the assigned task to other participants in proximity using a decentralized task distribution method (Eisenman, Lane, & Campbell, 2008). Participants who receive a task re-assignment request then verify if they can complete the task.

Finally, an emerging approach in participant recruitment is to leverage social network sites for calculating reputation as well as to expand recruitment of participants. For example, in addition to considering the quality of collected data, Amintoosi and Kanhere (2014) computed participants' trustworthiness level within a social network using a PageRank algorithm. The score of data quality and of the social network are combined to obtain a final reputation score. X. Hu et al (X. Hu et al., 2013), on the other hand, integrated crowdsourcing platform with social network sites to expand the scope of participation, ease the dissemination of data collection results, and facilitate user interactions through the interface participants have been familiar with. The results showed that integration of social network sites was efficient and required low communication overhead on mobile devices. Finally, Crowley et al (2014) also explored the use of social network sites for sending mobile crowdsourcing tasks. They showed that user's responsiveness is affected by user attributes such as device type, the rate of messaging, time of day, the social network sites being used, tie strength and path length. They also suggested researchers consider attributes of both users and the content of the request messages when using these sites for sending a sensing task.

2.2.4 Mobile Crowdsensing/sourcing Platforms and Campaigns

A number of platforms and campaigns have been built for deploying mobile crowdsensing tasks. Sensr (S. Kim, Mankoff, & Paulos, 2013) is an authoring environment claimed for enabling researchers without technical skills to build a data collection tool for citizen science. The objective of the system is to facilitate collaboration between researchers and participants through a campaign model. That is, people who seek data can author a campaign on the Sensr site. Participants interested in contributing the data can subscribe to the campaign and provide requested data.

D'Hondt et al. (2014) built a campaign providing orchestration support for participatory campaigns to achieve campaign quality, and automation of campaign to achieve scalability. The campaign framework can automate campaign definition, monitoring, and orchestration. APISENSE (Inria, n.d.) is a platform allowing researchers to recruit volunteers and to build and deploy crowdsensing applications for collecting sensor data. The platform, however, requires an invitation code to leverage the service. MOSDEN (Jayaraman, Perera, Georgakopoulos, & Zaslavsky, 2013) is a mobile sensing framework specifically supporting opportunistic sensing. It claims to be efficient in its separation of data collection, data processing, and data storage specific to an application. It provides a platform allowing data requesters to distribute opportunistic sensing applications. Very recently, an emerging platform called CrowdSignals.io claims to create the largest set of longitudinal data set collected from smartphones and smartwatches to data requesters who join the community. The platform uses a crowdfunding approach (Belleflamme, Lambert, & Schwienbacher, 2014) to sponsor a large pool of Android smartphone users in the United States to contribute their own smartphone data to the community. The community has obtained endorsements by a group of researchers interested in obtaining these data. However, this platform does not aim to support data requesters in deploying their own projects. It also does not support data requesters in requesting a specific type of data on the platform.

2.3 Mobile Data Collection Systems

Numerous mobile data collection systems have been developed for different purposes. It is a challenging to classify these systems because many of these systems, introduced for different research goals, have very similar sensing capability and functionality. Some systems are even built to serve as a generic tool and are not tailored to a specific application. In this section, I classify these systems into two categories. The first category is systems aimed at supporting

mobile crowdsensing. The second category is systems supporting context-awareness, including systems for behavioral research and for generic use. It should be noted that such a classification is arbitrary, and it is likely that systems in one category also satisfy the purpose proposed in the other category. However, given that one of the contributions of the thesis is an implementation of a mobile collection tool called Minuku that supports sophisticated context-triggered data collection, systems classified as the second category are those able to support context-awareness and generating context-triggered questionnaires.

2.3.1 Mobile Data Collection Systems for Mobile Crowedsensing

Campaignr ((Joki, Burke, & Estrin, 2007) is an early work allowing programming sensor data collection for mobile crowdsesning. It is written for the Symbian system and uses XML to specify data collection tasks and parameters, aiming to provide data requesters without programming experience with the access to all sensors on the mobile phone. It has a main controller that connects everything together. Thus, adding any components to the tool will need to modify the main component, reducing its extensibility.

EpiCollect (Aanensen, Huntley, Feil, al-Own, & Spratt, 2009) is another system for participatory data collection. It is on the Android system and allows participants to create entries of data such as location and images to send to a database. The system, however, does not support continuous sensing. Each data record is entered one-by-one by participants. It has very limited configurability compared to other crowdsensing systems reviewed in this section.

PRISM (Das et al., 2010) is a system for Windows phones that allows developers to configure sensing tasks in an application, and to package their application as executable binaries to send to participants' mobile phones. The system requires programing experience in order to configure the sensing task.

Medusa (Ra, Liu, La Porta, & Govindan, 2012) is another system supporting programming sensor for crowdsensing. The authors developed an XML-based programming language called Med-Script to provide high-level abstractions for *stages* in crowd-sensing tasks and for *connections* to describe the flow through the stages. Stages include: recruiting, task request, uploading, and data curation. The idea of staging and connecting is similar to trigger-action used in many tools like MyExperience (Froehlich, Chen, Consolvo, Harrison, & Landay, 2007b). However, it is very specific to the workflow of a crowdsensing task and thus is hard to apply to other types of research projects.

2.3.2 Mobile Data Collection Systems Supporting Context-Awareness

The second category is systems supporting context-awareness. Context ToolKit (Salber, Dey, & Abowd, 1999) is probably the earliest system that provides a detailed illustration of a software architecture of a context-aware system and allows developers to build mobile context-aware applications using context *widgets*. ContextPhone (Raento, Oulasvirta, Petit, & Toivonen, 2005) is developed with a similar aim, supporting developers to leverage the system to create context-aware applications, but it has a more complete support of context logging, and emphasizes a separation of context logging and storage from applications that react to contextual information, which, they argue, facilitates the construction of context-aware applications.

Context-awareness is essential to triggering context-based actions, such as prompting users a questionnaire. This feature is highly useful for interacting with participants in specific contextual conditions, including obtaining their subjective and qualitative inputs, usually regarded as a signal-contingent or an event-contingent experience sampling method (Reis & Gable, 2000; Reis & Wheeler, 1991). Numerous systems supporting context-triggered ESM have been developed. An early tool that has this capability is CASE (Intille, Rondoni, Kukla,

Ancona, & Bao, 2003). However, one major drawback of CASE is that it takes over the entire device and cannot run along with other applications, making it impractical to use it for conducting studies on users' mobile phones. Subsequent systems have addressed this issue and offer greater flexibility for dynamic triggers and configurable actions. Systems that have been mentioned include Momento (Carter et al., 2007) and MyExperience (Froehlich et al., 2007a). While the former is developed mainly for evaluating an early prototype of a context-aware system *in situ*, the latter records and monitors various contextual data, allowing researchers to define triggers that invoke actions when a specified contextual condition occurs. Despite that MyExperience has provided sufficient types of context information to collect and to monitor for triggering actions, its major limitation is that it supports a limited set of sampling strategies and ways to trigger an action, which might be insufficient for studies that need to execute actions in very specific conditions. In addition, MyExperience requires researchers to script inside configuration. This may become a barrier for researches without programming experience from using the tool. SocioXensor (Ter Hofte, 2007) is published in the same year as MyExperience, with similar features, including allowing researchers to configure types of data to collect and triggering ESM questionnaires. However, it only supports triggering questionnaires and requires more programing experience than MyExperience does.

Seo et al. (2011) particularly aimed to provide more functionalities for supporting ESM studies, with a focus on a web server, including tracking, reviewing and analyzing log data uploaded to the server and allowing the ESM tool to directly download the configuration from the server. However, probably due to the primary focus on ESM studies, its mobile phone client is only limited to conducting ESM studies but does not collect sensor data. Gerken et al. (2010), similarly, enhances web features such as remotely sending questions from the

server in real time. Additionally, it is claimed to have a more user-friendly interface for reducing users' burden in responding to an ESM questionnaire.

iEpi (Hashemian et al., 2012) is developed with an aim to support researchers and medical practitioners in monitoring health and treatment compliance. It further enhances the tool with a data analysis suit. ohmage (Ramanathan et al., 2012), a successor of the AndWellness system (Hicks et al., 2010), is claimed to be developed based on the feedback from hundreds of researchers behavioral and technology researchers, focus group participants, and end users. The novel key features of the system distinct from the aforementioned systems is the visual feedback of data on the phone, in addition to the provision of data visualization on the web interface.

More recently, Storyteller (Benjamin Poppinga, Oehmcke, Heuten, & Boll, 2013) is developed for enabling quick creation of storytelling for getting long *in situ* responses, of which the aim is to generate more accurate and substantial qualitative input from users. Psychlog (Gaggioli et al., 2013), on the other hand, is developed particularly for collecting psychological, physiological, and activity information for mental health research. This focus is enabled by combining self-reports and heart rate and activity information from a wireless electrocardiogram equipped with a three-axial accelerometer.

Instead of developing a standalone application, Lathia et al. (Lathia, Rachuri, Mascolo, & Roussos, 2013) built a set of libraries to collect, store, transfer, and query sensor data, as well as the capability to trigger time- and sensor-based notifications. The libraries separate sensing, data management, from triggering and thus allows researchers to import only the libraries they need (e.g. only background logging). However, as a set of libraries, researchers need experience in Android programming in order to utilize them for their studies. It is noteworthy

that the Funf, another open sensing framework[4], also provides open source sensing libraries, yet it does not support a context-triggered framework.

All of the aforementioned tools for collecting contextual data and supporting context triggered questionnaires have advanced the field of behavioral contextual data collection. However, there has not been a tool incorporating both features of mobile crowdsensing and context triggered ESM for researchers to use for conducting different styles of data collecting methods. In addition, these tools are still insufficient for conducting more sophisticatedly conditioned data collection because of the lack of support for a complex scheduling capability and a flexible trigger mechanism. In Chapter 6, I introduce a mobile tool called Minuku that equips these features.

Finally, researchers have sought to identify and investigate methodological challenges and issues of conducting ESM and electronic diaries, and suggest features that future ESM tools should possess. The key topics including user burden and compliance (Hufford, 2007; Morren, Dulmen, Ouwerkerk, & Bensing, 2009; Palmblad & Tiplady, 2004; Stone, Shiffman, Schwartz, Broderick, & Hufford, 2003), reducing user interruption (Ho & Intille, 2005; Intille, 2007; Pejovic & Musolesi, 2014a), and user recall bias (Cerin, Szabo, & Williams, 2001; Csikszentmihalyi & Larson, 1987; Hufford, 2007; Hufford, Shiffman, Paty, & Stone, 2001; Stone, Bachrach, Jobe, Kurtzman, & Cain, 1999; Vice-Chair, Pittsburgh, Institute, & Institute, 2007). The Minuku system addresses a portion of these highlighted challenges and suggestions.

[4] http://funf.org/

2.4 Collecting Annotations on Behavioral and Contextual Data

In addition to collecting behavior and contextual data, a number of systems seek to focus on collecting annotations. One class of these systems is to design for engaging users collecting the data in adding annotations. An important assumption behind many of these systems is that *only users who provide the data can correctly interpret the data and can provide accurate annotations*. The other class, in contrast, is to recruit users to review already collected data and then add annotations following a certain guideline. The assumption behind this system is that even the users who do not provide the data can accurately observe and interpret the behaviors in the data, and then add acceptably accurate annotations. Research projects belonging to the second class mainly take advantages of web services such as Mechanical Turk[5] for recruiting participants to annotate data, where the data is usually video clips (Lasecki, Song, Kautz, & Bigham, 2013). Literature of this class of systems is less related to the thesis and thus is not reviewed in the section below.

2.4.1 Collecting Annotations From Users Who Provide the data

Acquiring annotation is a common while vital activity in activity recognition that uses supervised machine learning algorithm (Kotsiantis, 2007). When using a supervised machine learning algorithm to train a model to learn features of captured activity data, annotations, including *ground truth labels*, are necessary inputs for training the model to learn associations between the labels and the features of the activity data. Because the accuracy of annotations directly affects the accuracy and reliability of the resulting activity recognition learning model, and that sometimes researchers may desire additional information about the

[5] https://www.mturk.com

activity data, a number of researchers have sought to contribute to collecting accurate and reliable annotations.

One focus on obtaining reliable annotation is to leverage visualization or video to allow users to review data for helping them recall. Most of these systems adopt a *post hoc* style, i.e. a visualization or a video is provided after activity data or the video is captured. Video recording is often used because it offers rich information about a person's activity and context within an environment, making it well suited to the application of behavior monitoring. It also does not require researchers to implement an additional visualization of the collected data. For example, DANTE (Cruciani et al., 2011) adopts a pair of cameras to monitor the movements of objects within a smartphone, and interprets the position and orientation of any object that is tagged with a marker. Videos are then reviewed frame-by-frame alongside the sensor data by users and are marked annotations post hoc. By using the video for verifying sensor data, the authors argued that DANTE reduced the amount of time required for annotation task by more than 45% compared to without using it.

CRAFT(Nazneen et al., 2012) also leverages video for adding annotations. But in addition to *post hoc* annotation, it also supports *in situ* annotation. Specifically, CRAFT is designed for capturing data about problem behaviors of children with development disabilities. While parents review videos and flag problem behaviors *in situ*, behavioral analysts annotate the videos *post hoc* and compare their annotations with the parents' to find agreements and disagreements. One highlighted weakness of this video-based annotation, however, is that the vast amount of videos requires extensive storage and a large number of hours to review the entire video footage. Another video-based system proposed is called MAVIS (Hunter, Donnelly, Finlay, Moore, & Booth, 2013), a mobile tool asserted to support *in situ* annotation in the home environment. However, unlike

CRAFT and DANTE, both of which initiate continuous video recording at all time, MAVIS requires users to manually record videos, a participatory style of data capture. In addition, the authors also propose a visualization of sensor streams called VISAVE, which is for complementing the MAVIS tool. Unfortunately, the two systems have not been implemented.

Another line of work is to ease the annotating process using voice recording, including (Harada et al., 2008) and (J. Y. Xu, Pottie, & Kaiser, 2013). Since an activity classifier would need a textual label, both research works have used speech recognition to automatically recognize spoken labels from recorded speech, and allow users to verify and correct labels to ensure accuracy. One challenge of this approach is that the speech might still be ongoing during transitions between activities. As a result, while a system recognizes labels from speech, it should also recognize transitions between activities to apply labels to appropriate segments of an activity trace.

Finally, recently Cleland et al., (2014) conducted an experiment to compare three protocols for acquiring ground truth labels, each of which specified how users need to record and annotate their activity. The three compared protocols were: structured, semi-structured, and free-living protocols. The free-living protocol adopts a context triggered *in situ* annotating approach. In this condition, users are prompted to annotate their activity by choosing an activity label when they are detected to be at the transition from standing after complete moving. In the structured condition, users are asked to carry out a number of activities for a period of time in the lab with specific instructions, where an observer keeps the timings of each activity and label the activity. In the semi-structured condition, users are not instructed how and for how long an activity should be carried out. An observer follows the users while they carry out the activities and label on the phone. The results of the experiment suggested that the accuracy of the labels that

were obtained through prompting users using smartphone were similar to the accuracy of the labels obtained in structured lab-based experiments. However, the authors only analyzed the accuracy of labels obtained from each condition but did not examine and compare the recorded activity data, which may show a different pattern.

2.5 Mobile Interruptibility, Receptivity, and Opportune Moments

Identifying opportune moments for sending annotation task requests to mobile users is an important as well as a promising direction to pursue. The Oxford Dictionary defines opportune as *well-chosen or particularly favorable or appropriate*. This definition implies that an opportune moment is not only a moment at which users are interruptible, but also a moment that users particularly favor and think is appropriate. Fischer in his Ph.D. thesis gave a similar definition. He defined that *a moment is opportune for a particular interruption if the participant is receptive to that interruption* (J. Fischer, 2011, p 81)

As mentioned earlier, it has been argued in much literature of ESM that using a context-aware computing technique to identify appropriate moments for triggering ESM questionnaires can help reduce the perceived burden of interruptions from mobile phones, thus improving user compliance and response rate (Ho & Intille, 2005; Intille, 2007; Pejovic & Musolesi, 2014a). Compared to responding to a questionnaire, recording and annotating personal behavioral and contextual data is likely to entail a higher burden and time, especially when using a participatory sensing approach, where actions needing to be performed occur beyond interacting with the phone (e.g. performing physical activities). Finding moments where participants are receptive to a request for collecting their personal and contextual data, thus, is a vital question to address in order to reliably, efficiently, and effectively acquire data as well as annotations from participants.

Literature in this space includes research on interruptibility, availability, attentiveness, responsiveness, and receptivity. Generally speaking, in HCI and communication literature, availability, attentiveness, and responsiveness are concepts used mainly for person-to-person communication, whereas interruptibility and receptivity are used more for interrupting users with a particular task. However, it should be noted that there has not been a clear distinction among many of these concepts, and that there has not been a common known way in which these terms are used and actually means in literature. Two meta-analysis papers of interruptibility literature (Sarter, 2013; Turner, Allen, & Whitaker, 2015), including a very recently published one, have suggested a challenge of defining and clarifying the notion of interruptibility. The reason is that these concepts are often mixed and interchangeably used in literature. In this thesis, I tend to use *receptivity* instead of *interruptibility* in the context of collecting behavioral and contextual data. The reason is that the former more precisely illustrate participants' willingness to accept a task request, which covers both attending to and responding to a task. These two concepts are important to consider to assess participants' reputation in contributing data. interruptibility, in contrast, does not necessarily imply whether or not participants would respond to a task.

2.5.1.1 Interruptibility, Availability, and Receptivity In the Workspace and Home Context

Early research in interruptibility focused on workspace communication, typically in an office setting on a desktop. Hudson et al. (Fetter, Seifert, & Gross, 2011a) conducted an ESM study to identify strong indicators for availability in the workspace. They concluded that the presence of speech (i.e., already being engaged in conversation with someone else) strongly correlated with being

unavailable for new interaction. Identifying sensors for availability indicators in the work context led to sophisticated models for predicting interruptibility and availability based on on-line calendar, readily available computer activity, and sensor information (Avrahami, Fussell, & Hudson, 2008; Begole, Matsakis, & Tang, 2004; Danninger, Kluge, & Stiefelhagen, 2006a; Fetter et al., 2011a; Fogarty et al., 2005; Fogarty, Lai, & Christensen, 2004; Horvitz, Koch, Kadie, & Jacobs, 2002). However, one limitation of using a predictive model to predict responsiveness to instant messages is that the predictive model built would be unaware of the content of the message (Avrahami et al., 2008). As the result, the model is likely to misinterpret, for example, a message indicating unavailability (e.g. "being busy now, talk to you later") as the recipient being available for a chat.

Research has also examined interruptibility and availability in the home environment. For example, Nagel et al. (Nagel, Hudson, & Abowd, 2004) investigated predictors of availability in the home environment using ESM. Unlike the previous studies for the workspace setting, the presence of speech is not a strong predictor of unavailability, and being alone also does not indicate being available. In their study participants were found most likely to be available when they were in and around the kitchen.

2.5.1.2 Interruptibility, Availability, and Receptivity On Mobile Phones

In recent years, interruptibility and receptivity research has begun to extend to mobile platforms, primarily the mobile phones. The advent of smartphones brings computer-mediated communication (CMC) into more diverse and unpredictable environments, making it more challenging to accurately interpret and predict interruptibility and receptivity to communication requests. After all, showing that users are using a mobile phone does not directly indicate the users' current

context and environment (e.g. at work, at home, or on the go), which may be influential on communication availability. In addition, *mobile* also implies possible rapid changes in users' current physical and social context, which in turn, can lead to different interruptibility and receptivity. Fortunately, most modern mobile smartphones have sensing capabilities, which afford new kinds of awareness and contextual information to use for sharing context and for predicting interruptibility.

Numerous research studies have explored sharing awareness and context information on mobile phones between communication senders and recipients to signal each other's communication availability. For example, Ljungstrand (2001) identified a need for sharing awareness of each others' context among mobile users to help them know the availability of the counterpart for accepting a phone call. Bentley & Metcalf (2009) showed how sharing mobile presence information—whether a mobile user is moving or not is enough to help mobile users coordinate activities. Similarly, ContextPhone (Raento et al., 2005) showed an application that allows mobile users to share awareness for the purpose of coordinating communication activity. Schmidt et al. (2000) also explored sharing context information to prevent inappropriate interruption, and De Guzman et al. (2007) investigated contextual information that helps a caller decide when to initiate a call. Mihalic & Tscheligi (Mihalic & Tscheligi, 2007) further explored how relationship type, mood, and communication channel and content affects how mobile users would like to be notified of a communication request on mobile phones, as Grandhi, et al., (2008) also showed that seeing who is calling is a important factor for mobile users to determine whether to accept the call. However, their suggested level of obtrusiveness of communication notifications was overridden by the participants 44% of the time. This result indicates that predictions of availability from sensor data was inaccurate at the time.

More recent efforts have been made to leverage context information collected on the phone for predicting mobile users' interruptibility and opportune moments for receiving different types of communication requests, including instant messages and phone calls. For example, Rosenthal et al. (2011) usesd ESM to acquire training data to develop a model for predicting phone call interruptibility that involves automatically silencing the phone when the user is uninterruptible. Similarly, Pielot (2014) built a model to predict whether mobile users would accept call using features including the time since the last call, the time since the last ringer mode change, or the device poster. They also (Pielot et al., 2014a) built a model to predict mobile users' attentiveness to instant messages using features including user interaction on the notification center, screen activity, ringer mode, and sensors. To my best knowledge in the literature, however, thus far there has not been research successfully developing a model accurately predicting mobile users' responsiveness to incoming communication.

The last but not the least relevant literature to finding opportune moments for delivering tasks is research on how mobile users attend to notifications on the phone, as attending to notifications is the first step of knowing that a task is received. Most of the research on mobile notifications are relatively new and are on Android phones. This is perhaps in recent years only the Android platform allows monitoring actions on notifications of the phone. For example, Pielot et al. (2014) adopted an ESM to study how mobile users deal with mobile phone notifications. They suggest that mobile users attend to notifications typically within several minutes regardless of the ringer modes of the phone (i.e. silent, only vibrate, sound and vibrate). Alireza et al.(2014a), on the other hand, conducted the first large-scale mobile phone notification study from more than 40,000 mobile users. Their results do not show *how* mobile users attend to notifications, but instead, reveal that mobile users value notifications from messaging apps and notifications including information about people and events.

While nowadays mobile users are likely to receive many, sometimes an overwhelming number of, notifications, it is important to consider how mobile users might prioritize reading different notifications, including the prompt notification sent from researchers. After all, overly dealing with notifications is likely to make mobile users feel stressed (Yoon, Lee, Lee, & Lee, 2014) which could lower their willingness to perform a requested task.

2.5.1.3 Exploring Opportune Moments for Delivering Notifications and Questionnaires

Finally, the last section reviews research in finding opportune moments for sending notifications. As just mentioned, finding such moments help researchers deliver important information to study participants or to conduct intervention more effectively.

Early research in this space has explored when to deliver notifications is least obtrusive and interruptive. Conducting in a lab-setting, McFarlane (2002) explored four different strategies of interrupting users performing a primary task (playing a handheld computer game) using a secondary task (a matching problem). The four strategies being compared in the study were: *immediately*, *scheduled*—on a regular basis, *negotiated*— users controlling when they would handle interruptions, and *mediated*— dynamically calculating a simple function of users' workload that measured how many certain objects in the game were currently visible on the screen. The study results showed that computer-identified-best-moments did not work as well as the negotiated method, suggesting that user determined moments more accurately reflect their favored time to be interrupted. This finding indicates that it may be worthwhile to allow mobile users to negotiate prompted time for receiving a task request. However, it should be noted that a negotiation option is available when a notification has been received and

attended to. Users are likely to be unavailable for acting on negotiation when they are not interruptibile. Negotiation also requires effort from users. In addition, the primary task in this experiment was a continuous game that demands high attention. Thus, the results may not apply to finding opportune moments within users' other daily life activities. Furthermore, a lab-based experiment has been argued for the lack of ecological validity. Whether the negotiating method is more effective than computer-mediated moments thus need to be reexamined in the real life setting.

Iqbal & Bailey (2008) conducted a similar lab experiment. Iqbal & Bailey developed a notification management system called OASIS to explore the effect of scheduling notifications at breakpoints between tasks detected by a machine learning model. The tasks in this experiment were designed to simulate what participants would do in practice in the real life, including diagram editing and programing. Their results showed that scheduling notifications at breakpoints reduced user frustration and reaction time compared to presenting them notifications immediately. While the results of this study are also promising, the results are still based on a lab experiment subject to the ecological validity issue. In addition, participants were exposed to notifications on a desktop computer, which was essentially different from them managing notifications on mobile phones in diverse environments. Breakpoints on mobile phones would be hard to determine because a large number of mobile phone use are micro-usage—less than several minutes, making the mobile phone mostly being in an idle state, compared to the programming and diagram editing activities (Ferreira, Gonçalves, Kostakos, Barkhuus, & Dey, 2014).

More recent research started to explore opportune moments on mobile phones. Fischer et al. (2011), for example, suggested that at endings of episodes of making calls and of receiving SMS indicate the breakpoints on mobile phones.

Notifications sent at these moments would be dealt with significantly more quickly than a random baseline condition. Poppinga et al. (2014) investigated mobile users' responding and ignoring notifications with 79 mobile users. From the data they collected, they developed a decision tree model for predicting opportune moments on mobile phones to deliver notifications, with 77% accuracy. The research suggested that good indicators of opportune moments for sending notifications include phone position (people holding the phone), time in a day, and location. Similarly, Pejovic et al, (2014b) developed an intelligent prompting mechanism called InterruptMe for exploring opportune moments for delivering questionnaires. They used ESM to deliver questionnaires asking mobile users' emotional states, whether they felt bored, and whether it was a good moment to interrupt. They used collected responses to train a learning model for finding opportune moments for delivering questionnaires. Features that were found as good indicators of opportune moments included physical activity, location, time of a day, and engagement. Sarker et al. (Sarker et al., 2014) also used ESM to train a learning model to predict opportune moments for answering an ESM questionnaire. The research suggested that location, emotion, physical activity, time of a day, and day of a week play an important role in predicting availability for answering an ESM questionnaire, thus arguing that these are the opportune moments where mobile users would be available for real-time intervention. Smith et al. (Smith & Dulay, 2014), in contrast, focused on opportune moments for making phone calls. The research compared performance of different marching learning algorithms in predicting when mobile users would think a call is disruptive. Availability for answering a call is inferred from users' action of answering, declining, ignoring, or choosing an *silence answer* option—a new option of a new call interface provided by the research that tells the phone to answer but silence the call in the future. The research suggests time of the incoming call and location are two predictors.

At a higher level, these research studies suggest a consistent result: physical activity, emotional states, location, and time of a day are good indicators of availability for answering a questionnaire. In addition, features such as phone position and users' actions on the phone activities are also strong indicator of availability for dealing with notifications. One remaining question, however, is the applicability of the findings and suggested predictive features to finding opportune moments for delivering tasks of collecting personal behavioral and contextual data. It should be noted that an essential idea of receptivity is people' willingness to perform a task, where the willingness may depend on the content of the task. Being willing to answer a questionnaire does not necessarily mean being willing to perform the activity to be recoded and annotated.

As discussed earlier, collecting personal behavioral and contextual data are likely to require higher cost from mobile users to perform the activity to be recorded and annotated. The cost may be due to inappropriateness to perform the activity in their current context, or due to the higher burden to perform the activity than to answer a questionnaire. Due to the applicability issue, future research is needed to explore the contextual factors predictive to mobile users' opportune moments for collecting personal behavioral and contextual data.

Chapter 3 Using Capture-And-Playback to Support Prototyping, Testing, and Evaluation of Context-Aware Applications: Findings and Lessons Learned

3.1 Introduction

As the venues for human-computer interaction move beyond office walls and into places like homes, health clinics, and public spaces, designers face new challenges in understanding users' existing practices and needs as well as in designing appropriate application. These challenges mainly lie in the difficulty of anticipating how the application will behave in different contexts of use. This is especially true for context-aware applications—i.e., applications that adapt their behavior to sensed aspects of users' behavior and/or environmental conditions. In such cases, designers not only need to understand the contexts of use but also need to work out the ways that the application will respond to changing contextual conditions. Knowing how an application will behave in the context of use is not just a problem for designers who need to decide what features the application needs to provide and how those features ought to be presented to users; the software developers who are charged with making the software work reliably in the field are also at a disadvantage because it is difficult to recreate the range of inputs that the software will have to deal with.

One solution to this challenge is to lower the barrier to deploying context-aware applications in order to more rapidly move prototypes into the field for obtaining real-world experience and feedback. A number of prior design tools have sought to provide better ways of "bringing the lab into the field" through support for rapid prototyping of pervasive computing applications (A. Fouse, Weibel,

Hutchins, & Hollan, 2011), field testing lo-fi prototype (Carter et al., 2007), and Wizard-of-Oz support for early stage tests involving context-awareness (Li et al., 2004; Li & Landay, 2008; MacIntyre et al., 2004). However, despite the availability of these tools, the bulk of the work involved in designing, developing, and evaluating software of any kind does not happen in "the field," but rather in offices, cubicles, studios, and labs where designers and developers are involved in the reflective activities of design-build-test on a daily basis. Tools that help "bringing the field into the lab" by allowing designers and developers to more easily recreate contextual conditions at design time that their systems will have to face at runtime, then, is an important complement to the existing ubicomp toolbox.

As a step in this direction, Newman et al. (2010) built a capture-and-playback (C&P) system called RePlay that allows designers and developers to capture contextual data representing user behavior in the field, and to play back the captured data traces to help with rapid prototyping, testing, and evaluation of context-aware applications. Specifically, RePlay provides a set of features to support the C&P process in addition to the core capability of capture and playback. The features include: a) synthesizing captured traces into "Episodes" involving multiple individual behavioral traces that simulate "scenarios" a context-aware application is anticipated to encounter in the field, b) a library providing access to all captured traces organized by the users of the system, and c) transforming data to create permutations of data that might be useful for particular purposes. However, while these features appear to offer advantages for developing context-aware applications, questions regarding their effectiveness remain. To date, little work has investigated two research questions crucial to context-aware application development: a) what challenges designers and developers would encounter when using a C&P approach and a tool to design and develop context-aware applications; and b) what kind of support a C&P tool

should provide to address the challenges to make the C&P approach more effective. Our goal is to fill the gap by answering these two research questions and to inform the design space for a C&P tool.

To advance this goal, we evaluate the features of RePlay for prototyping, testing, and evaluating a number of context-aware applications. Specifically, we followed two directions to accomplish our goals. In the first direction, we undertook two design projects featuring different interaction modalities and different design concerns that made substantial use of captured data. Based upon the interaction design lifecycle model described by Rogers, Sharp, and Preece (Rogers et al., 2011), we planned and executed one full circuit of the interaction design lifecycle for each project and used an improved version of RePlay to prototype, test, and evaluate the two systems. Our goals for these two case studies were to reflect on our own experiences in exploring the benefits of and the challenges in using a C&P approach and tool to design and develop location-aware systems. Throughout our efforts in this direction, we showed that using C&P approach and tool is beneficial to prototyping, testing, and evaluation of the two context-aware systems we built, at least in: helping answering design questions; examining design alternatives; testing features and algorithms; and creating realistic conditions for engaging participants in system evaluation.

In the second direction, we aimed to inform the design space of a C&P tool through investigating developers' needs and behaviors in using RePlay to test and evaluate a location-aware application. Based on the study results, we identify three important activities a C&P tool should support in testing and evaluating context-aware systems—selecting examples, modifying data, and control playback during iterative testing. We then improved RePlay with an aim to support these activities and evaluated the effectiveness of the new system in supporting these activities.

Throughout these two directions, we summarize three major challenges designers and developers encountered in using a C&P approach and a tool that researchers need to address: a) possessing data needed for various development activities; b) knowing what data is available for use for different development activities; and c) selecting and creating suitable examples for playback among a large amount of captured data. To address these challenges, we argue that a platform that equips the features of CaPla and supports accessing, sharing, and requesting contextual and behavioral data is needed. We believe such a platform can more effectively support prototyping, testing, and evaluating context-aware applications.

Below, we first describe the features of the improved RePlay system we evaluated throughout the two directions.

The RePlay System

RePlay (Newman et al., 2010) is a system for capturing, organizing, transforming, and playing back sensor traces representing user behavior and application context during development. It assumes a system architecture wherein context acquisition and processing are separated from interactive clients, which is a common high-level pattern for context-aware systems (e.g., (Bardram, 2005; Dey, Abowd, & Salber, 2001; Hong & Landay, 2004; Winograd, 2001). RePlay inserts itself into the system as a "context service," using previously captured service outputs to masquerade as the services that will ultimately provide context data to the clients and other services. This means that clients can remain unaware of the source of the context data they receive, allowing a smooth transition between using RePlay's captured data and live sensor feeds as the development process progresses. In our implementation, RePlay communicates with clients via Whereabouts (Ackerman et al., 2009), a privacy-preserving XML-based Blackboard system.

Leveraging this architecture, RePlay provides a set of mechanisms that allow developers to work effectively with captured data and integrate it into the development process:

- *Capture Probes* are lightweight services that can record various sensor traces. They come in two types. Embedded Probes are fixed sensors that are temporarily installed in an environment to sense the activity of its occupants. Examples we have built include Bluetooth, Video, and Noise Level Probes that run on a small Linux-based PC with appropriate hardware (i.e., Bluetooth radio, webcam, and microphone respectively) and can log traces of activity in the Probe's environment. Mobile Probes feature sensors that can be carried by target users to track aspects of their own behavior such. For example, our latest developed Android-based Mobile Probes, namely, Minuku, can capture location traces, modes of transportation (e.g. in a vehicle, biking, walking), Bluetooth sightings, network status, application usage, and a variety of sensor readings available on Android phones.

- *The Clip Library* stores Clips—sensor traces captured by the Capture Probes that represent user behaviors. Each Clip consists of a sequence of timestamped tuples, each of which contains the data from a single sensor reading.

- *Episodes* are collections of Clips that represent coherent traces of user behavior and/or contextual conditions. An Episode can include Clips representing multiple people; multiple streams associated with a particular person; or any combination of Clips and associated users that the developer deems useful for a given project. As an example, an Episode comprised of GPS Clips representing five users' movements on a Friday evening around

6pm might be labeled “Friends Meet for Happy Hour” to denote a scenario of importance to a location-based social networking application.

- *Transforms* are processing units that allow developers to manipulate Clip data to make it better fit anticipated usage situations. In our experiences with the early version of RePlay we rapidly learned that raw captures are not always immediately useful for design. Regions of interest may be too long or too short, or may not have captured particular events that the application would be expected to encounter in an eventual deployment. Examples of Transforms that we have found useful include the Delay Transform, which changes a Clip’s start time to align with other clips in an episode; the Dwell Transform which introduces an artificial pause in the data to lengthen the time that an observed behavior occurs; the Identity Transform, which changes the identity of the user associated with a particular Clip; and the GPS Noise Transform which introduces artificial noise into a GPS-based location trace in order to test degraded performance.

The data contained in Episodes can be played back and monitored via the RePlay UI. RePlay’s Player window controls the playback of Episode data, while the World State window gives an overview of the current state of the playback at each moment in time, providing a range of visualizations for different kinds of data. The Clip Library window provides access to all captured data and supports the creation and selection of Episodes. The improved RePlay UI is implemented in Adobe Flex and communicates with Whereabouts via the Java-based Replay Engine that manages communication, provides access to the Clip Library, and executes Transforms using a plug-in architecture that enables the addition of new Transforms.

Figure 3.1 The RePlay user interface consists of the World State window (A and B, upper right), the Player window (below the World State window, the Episode Library (A, center), and dialogs for previewing specific Clips (A, left) and editing track data (A, lower r

Note that the version of RePlay described in this paper adds several improvements to the previous version in (Newman et al., 2010). The new version, shown in Figure 3.1, differs from the previous one in several ways. First, the visual presentation and many details of the user interaction were overhauled as part of the process of porting RePlay from Java/SWT to Adobe Flex. In addition to aesthetic and usability improvements, RePlay was redesigned as a "sidebar" (i.e., a relatively small window running alongside one edge of the user's monitor) allowing it to better exist alongside other design and development tools than did its full-screen predecessor. Second, the Clip Library was redesigned and renamed to the Episode Library (the center of Figure 3.1) to give greater primacy to Episodes and to provide easier access to semantically relevant descriptions of

both Episodes and Clips. Third, additional control over playback was added with the addition of Repeat Play (the ability to play a set of Clips repeatedly without manually restarting) and Play Regions (adjustable markers on the timeline that allow the user to set particular start and end times). Finally, *Annotations* were added to allow users to anchor textual notes to particular timepoints within Clips and Episodes in order to more easily re-find events of particular importance. These improvements were added based on our early experience in using RePlay in developing several context-aware prototype we built. However, we only obtained preliminary and informal feedback internally and had not evaluated them in a more formal way. It is thus our aim in this paper to investigate how to better support the design and development of context- aware applications.

3.2 Research Goals and Approach

Our goal is to provide insight into how a C&P tool such as RePlay can help in context- aware systems development. In this chapter, we mainly focused on location-aware system because it is by far the most common type of context-aware system in current commercial development, and therefore the datatype of location we felt would be the most familiar to the developers. We see the design space for a C&P tool as large—potentially encompassing multiple tools for different categories of system development. In addition to standalone tools like RePlay, we see a continued role for C&P related features integrated into dedicated design tools like Topiary (Li et al., 2004) or Panoramic (Welbourne et al., 2010), as well as into IDEs for developing mobile phone systems such as Eclipse[6],

[6] https://eclipse.org/

Android Studio[7], Xcode[8], and Visual Studio[9] The work described in this paper, then, seeks to inform the design space for this class of tools by asking what features a C&P tool should equip to support different activities involved in the design and development process of context-aware applications. While it would be desirable to eventually deploy RePlay into realistic settings and observe how it is used by different development teams, there is much that can be learned through Case Studies—we researchers, as also potential users of RePlay, reflect on our own experience in using RePlay in developing real design projects, and User Studies—learning representative users' need and behaviors in using RePlay for testing and evaluating location aware applications. Below we talk in more details about the two directions.

3.2.1 Case Studies

The goal of these case studies is to reflect on our own experiences in exploring the benefits of, and the challenges in using a C&P approach and tool to design and develop location-aware systems. To advance this goal, we engaged in two substantial, realistic design projects featuring different interaction modalities and different design concerns, in which the ultimate system would feature location data prominently. The first case study is LoungeBoard. It is a proactive public display intended for installation in a community lounge that encourages social interaction among students, and shows information relevant to the students in the lounge. The second project is BusBuddy, an Android location aware application for tracking public buses and planning transit journeys. During the course of designing and developing two systems, we captured many hours of location traces

---

[7] http://developer.android.com/sdk/index.html

[8] https://developer.apple.com/xcode/

[9] https://www.visualstudio.com/en-us/visual-studio-homepage-vs.aspx

representing dozens of examples of user behavior and expected contextual conditions. We used RePlay to play back the captured location traces in subsequent prototyping and evaluation activities, though we used a variety of tools in addition to RePlay to capture, process, and organize the captured data.

As the projects progressed, we sought to follow "standard" interaction design practice as closely as possible, choosing specific methods appropriate to each project. Taking the "simple interaction design lifecycle model" described by Rogers, Sharp, and Preece (Rogers et al., 2011) as our reference point, we planned and executed one full circuit of the interaction design lifecycle for each of the two projects. That is, we engaged in establishing requirements, designing alternatives, prototyping, and evaluating each system, incorporating data capture and playback into the process wherever it appeared to be helpful. Throughout, we continually reflected on our own experiences to understand where we found benefits of C&P in different design and development activities, where we found it difficult, and what solutions we came up with to ease the difficulties and amplify the benefits. We carefully documented our experiences, lessons learned, and the insights we gained as we handled and made use of the data. While it is our responsibility to persuade the reader that our design process was reasonable and represents at least a minimally competent execution of a realistic design process, we will not strive to provide arguments or evidence that our designs are "good" according to any extrinsic metric of quality. Rather, our goal is to highlight junctures in each process where the use of C&P approach and tool helped each design activity.

3.2.2 User Studies and Continue System Improvement

Previous research is lacking in understanding the process of interacting with captured data during development as well as the challenges that developers encounter in using captured data in working with context-aware systems.

Understanding this process and improving the tools that support it is the focus of the user studies. To gain insight into the issues that developers would face in using a C&P approach and tool as part of development activities, we invited developers to participate in a user study to modify and improve a location-aware smartphone application we had built. The goals for the study were to understand how developers would interact with captured data while working on concrete tasks; to see how the features of RePlay helped or hindered developer' development tasks; and to identify additional features for C&P tools that would help developers work more effectively. Based on the results from the user study, we continued to improve RePlay to set out to build a comprehensive set of tools that would support different identified important development activities. The resulting toolset, called CaPla, a direct descendent of RePlay and TraceViz (Y. Chang, Hung, & Newman, 2012), is then evaluated by a follow-up user study to understand whether the added features did support testing and evaluation as anticipated.

3.3 Case Studies

3.3.1 Case Study 1: LoungeBoard

LoungeBoard is a proactive ambient display intended for installation in a student lounge. It runs on a large display and is continually refreshed with textual and graphical content presumed to be useful or interesting to the lounge's current occupants detected to be close to the display. The LoungeBoard project started as an exploration of the possibility of using public displays to augment our school's student lounge for master students. We were inspired by previous work in shared-space public displays (Izadi, Brignull, Rodden, Rogers, & Underwood, 2003) and proactive displays (Congleton, Ackerman, & Newman, 2008; McCarthy, Congleton, & Harper, n.d.), and saw an opportunity to develop a system that

could improve our environment while allowing us to carry out a realistic, substantial design project using a C&P approach.

3.3.1.1 Capturing Data

The process of establishing requirements for LoungeBoard was tightly coupled with the data capture process, with each activity informing the other. We conducted five hour-long field observation sessions in the lounge at different times of day, each time writing field notes describing the individual behaviors and social interactions that were taking place. The purpose of these observations was twofold. First, we sought to get a sense of the nature and diversity of activities that take place in the lounge, and to better understand the social environment. Second, we sought to develop our plans for capturing data by identifying different social situations that would be useful to capture and represent in our design process. We also conducted a few informal interviews with informants to understand why master students came to the lounge and what information they were often seeking while they were staying or waiting in the lounge.

Observing the ebb and flow of social interactions across different periods helped us identify specific scenarios that would be important for LoungeBoard to address. However, At this point, we were not sure what contextual data could be useful to the application; we decided initially to capture contextual data in the form of video. We made this decision not because we expected LoungeBoard to employ video-based sensing when ultimately deployed, but because video could unobtrusively capture rich contextual data with essentially no infrastructure requirements, allowing us to easily carry out the capture and giving us maximum flexibility to revisit and reinterpret the data later. Here we were influenced by previous projects that demonstrated the feasibility of using hand-coded video to

simulate sensors in the development of ubicomp systems (S. Consolvo, Arnstein, & Franza, 2002).

To produce reusable "sensor" traces, we used Lag Sequential Analysis (LSA, following (S. Consolvo et al., 2002) to manually code occurrence of contextual elements such as individual students' presence, location within the room, orientation, and pose (e.g., sitting, standing) that we thought potentially useful and technologically feasible for the team to develop in the future. However, the effort required to code the video at sufficient granularity (we used a "lag" of five seconds) was quite onerous, and the effort required would outweigh the benefits. This made us consider more carefully which elements of context would be most useful for the application, and to code selectively just for those elements.

On the other hand, from the informal interview we also learned that many students went to the lounge between classes for social interaction and taking rests. They also went to the lounge for waiting for buses or for meeting rooms to be available. These findings inspired us to develop a service that could trigger social interaction and meanwhile inform students about bus arrival times and availability status of meeting rooms. The findings also made us decide to capture bus location traces if later in the design phase we determined to include bus arrival time information in the system.

3.3.1.2 Designing Alternatives and Organizing Episodes

Based on our understanding of the social and physical environment of the lounge and the information master students in the lounge desired to know, which were encapsulated in our initial set of scenarios, we began to sketch different design ideas. We considered a wide range of system concepts, ranging from gesture-based group games to media space-like displays linking the lounge to similar

spaces in other buildings. We finally decided that the display should play a peripheral role in the lounge, not demanding attention but providing interesting and useful information at a glance for students. Moreover, we refined the notion of "interesting information" to mean information that provides material for conversation, as social encounters were among the most common and apparently desirable experiences that attracted students to the lounge. In addition, we refined the notion of "informative content" useful to the students, such as bus arrival times at a nearby bus stop and the availability status of meeting rooms down the hall from the lounge. Because the original purpose of the system was to facilitate interaction, we determined to set our initial focus on providing "interesting information."

This concept refinement led us to focusing on occupancy, the set of people currently in the lounge, as an essential contextual stream that would be helpful for selecting and displaying information interesting and relevant to the particular occupants of the lounge at any particular time. With this renewed focus, we drew on our prior observations to identify time periods where there might be different types of social arrangements, including both large and small crowds, and periods of high and low turnover (more and less frequent entrances and exits). After two failed attempts (the lounge was empty or nearly empty) we found that around noon, and before and after afternoon, were the time periods where more students would leave and enter the room. Evenings and morning would be good times to find smaller, less active crowds. We therefore captured five hours of data across three sessions and coded them for arrival and departure events, as we anticipated we would test LoungBoard with these conditions. This process was significantly faster than the initial, broad-based coding effort, and yielded a set of traces that promised to be useful going forward. Reviewing the five hours of coded data, we were able to find examples of all of the important dynamics that the LoungeBoard would need to handle.

Because of the care taken in planning and executing the data capture, fairly little work was needed for actually organizing the data. A team member wrote a small script for importing data from the spreadsheet containing the LSA codes into RePlay, which generated Episodes representing the changes in occupancy across each time period originally captured. These Episodes would have been too long (1-2 hours each) to be practical for subsequent phases, though, so we additionally extracted 24 Episodes representing key scenarios for use in prototyping.

In addition, because we had determined to show bus arrival and meeting room information on LoungeBoard, we also started collecting location traces of buses. At this point, we did not capture meeting room occupancy data using videos because compared to a student lounge, meeting room is relatively private space. We also did not generate any Episodes of bus traces because at this point it was not entirely clear what Episodes of bus traces would be useful. As a result, we chose to defer organizing and generating Episodes of bus traces to later phases.

3.3.1.3 Prototyping

Armed with high-level goals, initial sketches, and Episodes representing key scenarios, we set out to build a prototype. Our Flash-based prototype displayed two forms of informative content—bus arrival times and the meeting room availability—and a rotating set of interesting content intended to spark conversation among lounge occupants. The interesting content was chosen based on who was in the room as reported by RePlay, with the intent that such occupancy data would be eventually provided by context sensing services in a future deployment.

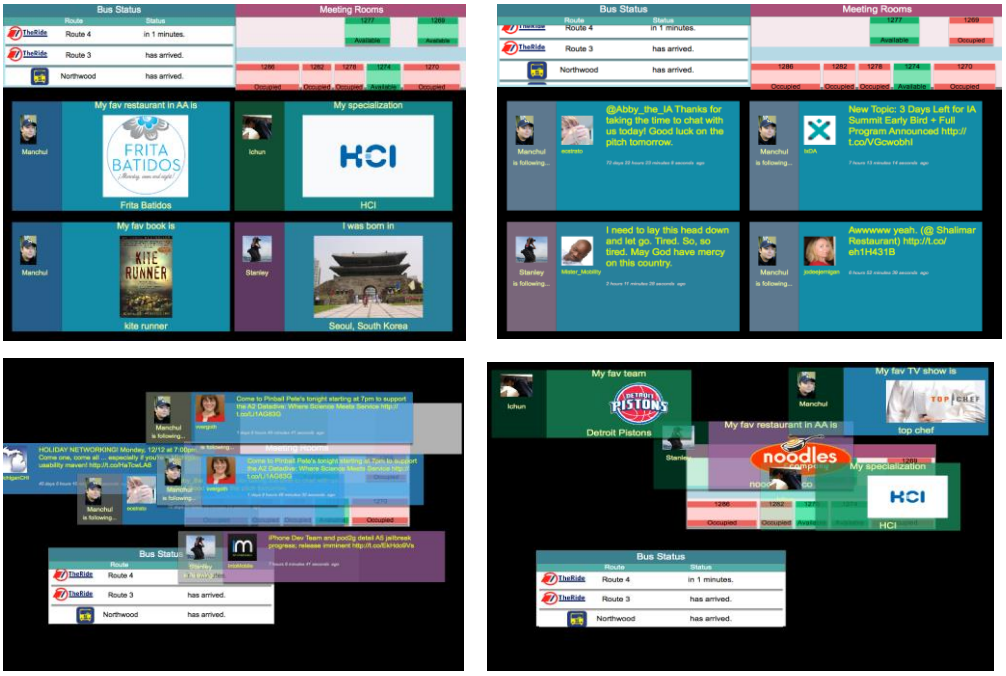


Figure 3.2 The LoungeBoard Interface. The screens on the top row show a "border" display style. The screens on the bottom row shows a "collage" display style.

The prototype was developed in conjunction with our plans for conducting user enactments (Davidoff, Lee, Dey, & Zimmerman, 2007), a technique for "identifying the overlap between observed needs and perceived needs" [p. 433]. User enactments expose potential users to variations on a design concept in the context of "invented scenes" that approximate the situations in which users would encounter the design after it was deployed. As such, developing variations of our prototype was critical, so our prototype varied along two dimensions: display style and content type.

We designed two display styles: a "collage" style and a "border" style (see the interface of LoungBoard is Figure 3.2). The "collage" style intermingled informative and interesting content in a random arrangement, with all content appearing at irregular intervals on the screen and floating for a period of time

before fading out. The “border” style displayed the informative content in a fixed location along the top of the screen, while interesting content rotated in a regular counter clockwise pattern through a panel occupying the bottom ¾ of the screen. Interesting content was in two types: a “water cooler” content type featured a subset of recent Tweets from Twitter users followed by each lounge occupant, whereas an “ice breaker” content type showed potentially interesting information about each occupant based on profile information they had submitted when signing up for an imagined LoungeBoard service.

As we developed the prototype, we repeatedly played through Episodes we had created earlier. Specifically, we played back ten Episodes representing different crowd sizes and patterns of arrivals and departures. This helped us identify issues with the prototype that would affect not only the user enactments study but also usage scenarios beyond what we would be able to test. We learned, for example, that our initial heuristics for rotating content in the collage style worked reasonably well for up to four occupants. But it became unreadable after that point. We also learned that the border style behaved poorly when there were frequent departures from the room, and realized that we needed to implement a more graceful way to “expire” exiting content when people left the display. In addition, we selected and played individual bus location traces to examine different presentation styles of bus arrival times, such as the format of showing arrival times and the frequency of bus location udpate. Observing occasional dramatic changes in the arrival time through the playback due to the noises contained in the location traces collected on mobile phones, we learned to adjust the frequency to smoothen the change. These issues could have been quite detrimental to the user experience of the LoungeBoard, yet we don’t believe we would have caught them if we hadn’t had easy access to the realistic and varied test data that we had captured earlier in the project.

3.3.1.4 Evaluation

For our user enactment study (Davidoff et al., 2007), we invited 8 participants to visit a simulated lounge in our lab and exposed each of them to four different vignettes that varied along four dimensions. The first two dimensions are the prototype dimensions described above: content type and display style. The other two dimensions are "ambient trigger" and "social situation." An ambient trigger was purposed to trigger a participant to perform an expected action. For example, LoungeBoard showing a targeted bus route going to arrive at the closet bus stop was a trigger for the participant to go catching the bus; showing meeting rooms being available was a trigger for occupying the meeting room. A social situation was simulating a situation involving social interactions (e.g. conversation) in the student lounge. In our study design, each vignette consisted of an ambient trigger and a social situation. We recruited two outgoing students as actors to pose as lounge occupants in each vignette. Both actors and participants were given a high-level description of each condition and system variation and were asked to "act out" a scenario involving the system (shown in Figure 3.3).

Among the four vignettes, two involved waiting for a bus and two involved grapping a meeting room. In two vignettes involving waiting for a bus, participants were told to catch the next #3 bus and were informed that it would take about one minute to walk to the stop. Roughly six minutes into the vignette, a route #3 bus was shown as "arriving in two minutes," at which time we expected the participant to get up and leave the lounge. Before then, participants could feel free to interact with the two actors. In vignettes involving grabbing a meeting room, participants were told that they had volunteered to get a first-come first-served meeting room for a group meeting happening soon. Initially all six rooms are shown as occupied, but about seven and a half minutes into the vignette, one of the meeting rooms became available, triggering the participant to vacate the lounge. Later on, we changed one vignette involving grabbing a meeting room to

Figure 3.3 Our design team used the user enactment technique to evaluate LoungeBoard. Actors recruited and participants were given a high-level description and "acted out" a scenario involving the system.

be more relaxing in order to make the participant less driven by the task of trying to getting a meeting room as soon as possible. That is, in the new meeting room vignette, LoungeBoard initially showed 5 available meetings rooms, which then became gradually occupied. The participant was free to hang out in the room until he or she needed to leave the lounge to get a meeting room before all of the rooms were occupied. After substituting the new vignette, participants were found more engaged in the conversation with the actors while also watching LoungeBoard more frequently.

Each vignette was associated with a different social situation, including different ways to initiate conversation with the participant, different initial settings, and different times at which the two actors arriving and leaving the room. These differences were designed to ensure variety and freshness across the four

vignettes. In the debrief after the study, all participants reported that the vignettes were realistic to them and were common activities in the lounge.

We used RePlay to play back Episodes to simulate bus status and room occupancy. For bus status, we played back four bus location traces as an Episode to simulate bus arrival events. The route #3 trace was trimmed in a way that it arrived at a nearby bus stop in 8 minutes. Other bus location traces only served as distractions and were not modified. For simulating meeting room occupancy, because we did not capture meeting room occupancy data, we created an Episode containing six manually produced Bluetooth signature traces for representing each of the six meeting rooms. However, to simulate a social situation reflected on the display—showing interesting content depending on who were present in the room—we had to improvise using the Wizard of Oz technique (Dahlback, Jonsson, & Ahrenberg, 1993) in order to avoid a situation where the participant perceived a dissonance between what was shown on the display and who were present in the room. Therefore, for each vignette, a member of the design team played the role of "wizard", who played back an Episode of three 15-minute continuous occupancy traces representing the participant and the two actors, during which he controlled the playback of each trace according to the condition of the vignette he observed.

After the user enactment study, most feedback we obtained from participants was related to personal comfort with the interesting content displayed on the screen, interface design, the usefulness of informative content, and future design direction. All participants regarded bus status and room availability highly useful and informative to them. A few participants suggested including more interactive elements such as providing sound alert or touch-based screen. In addition, while some participants indicated the need of including the reservation information of

meeting room on the display, others were concerned with the credibility and accuracy of the information source.

3.3.2 Case Study 2: BusBuddy

BusBuddy is an Android application that helped bus riders track buses' locations and expected time of arrival (ETA) while planning bus trips or waiting at bus stops. BusBuddy was aimed to build upon crowdsourced location sharing mechanism that was inspired by the recent bus checking tools such as the OneBusAway (Ferris, Watkins, & Borning, 2010), NextBus ("Nextbus," n.d.), and Tiramisu (Zimmerman et al., 2011b) and upon bus time schedule provided by a bus company if a location update from user were not available. The core idea of BusBuddy is that a bus waiter registered on the BusBuddy community is able to inquire the location of a bus that he or she monitors via the system. The inquiry is then directed to bus riders currently on the bus heading to the inquirer's location. If and only if a bus rider receives a location inquiry and choose to share his or her location to the system, would the inquirer, the bus waiter, receive the location of the bus rider, which presumably represents the bus's current location. In our case study, our design was focused on the interaction design from the inquirer's perspective.

3.3.2.1 Establishing Requirements and Capturing Data

In order to understand what information would be useful and desired for a bus waiter to see on the BusBuddy interface, we conducted informal interviews with five informants, three taking buses on a daily basis and two taking buses on occasions. We asked them about what bus-related information they desired to know when waiting for a bus, and to describe their prior experience and strategy of waiting for a bus. Focusing on these questions helped us build a sense of the bus-related information that would be useful to provide by BusBuddy. For

example, we learned that around the time during which a bus was anticipated about to come was a "critical time" for bus waiters to make the decision whether to wait or to select an alternate route. Being uncertain about whether an anticipated bus has left or not made it difficult for the waiters to make a decision. As a result, we decided that BusBuddy should provide bus location and ETA to assist bus waiters in making such a decision.

Data capture was undertaken in parallel with the interviews. Initially, we did not focus on specific bus routes and wanted to collect a wide range of routes. As the result, in addition to our own team members collecting bus location traces, we also recruited seven individuals who took the bus regularly to record their location when riding a bus. We provided these helpers a high-level capture guideline including a list of available GPS loggers they could use on Android phones and iPhone; when they should turn on and off location recording; and how to send recorded location traces to the design team. In the initial plan of data capture, we collected 46 bus location traces in seven different routes, including 25 routes collected by the helpers. The lengths of the collected traces varied; the longest location trace was 1 hour 31 minutes long, and the shortest trace was only 7 seconds long, which might be due to an accident during the recording. The data capture lasted one and ½ months.

3.3.2.2 Designing Alternatives and Organizing Episodes

Our initial design focuses on how we presented bus location and ETA to the closet stop on a map. We encapsulated identified user needs into three scenarios to illustrate how BusBuddy would be used, and explored different design ideas based upon these scenarios. We considered a wide range of design questions but at the end converged on five that we were interested in examining before the first round of testing and evaluation. For example, we decided to examine the

effectiveness of color-coding in distinguishing among various bus routes and examine how to utilize the limited interface to presenting bus icons, bus routes, and bus stops.

We first tried to synthesize Episodes based the initial three scenarios. To know which bus trace could be used, we observed the path of each captured trace through RePlay. However, we found a lack of bus traces for creating Episodes matching the created scenarios. This made us adopt a bottom-up approach to create Episodes. i.e. creating Episodes based on the data we possessed. In inspecting the content of traces, we tried to make the file name of each trace reflect its content, so that it was easier for us locate them when we wanted to create an Episode containing a particular encounter. In addition, when we found bus traces containing unanticipated movement or incidents (e.g. signal lost), we also modified the file name to reflect those incidents. Furthermore, when inspecting, we also attempted to think ahead: What would be good Episodes for prototyping and testing, respectively? How could I use this trace to create different kinds of Episode? How would the BusBuddy UI look like if we played back this "problematic" segment of the trace? We then tried to synthesize Episodes that contained conditions we regarded as interesting to evaluate BusBuddy, such as three different bus routes arriving at different corners of a building at the same time.

At this stage, we have attempted to generate Episodes that we anticipate to use for usability testing and prototyping. For the former, we aimed to find bus traces that would create a realistic use cases in a usability test. However, at the time we had not established any concrete usability test plan. Thus, this process was exploratory. For prototyping, in addition to the traces we identified earlier as "problematic" ones, which we considered useful for evaluating the prototype of BusBuddy, we also created a set of Episodes with randomly selected bus traces to

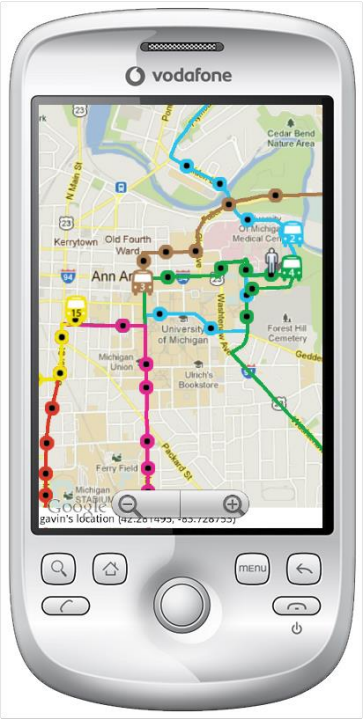


Figure 3.4 The interface of the BusBuddy prototype

reflect the unpredictable nature of the real world. We assumed that a random combination of traces might give us some surprise and helped us find unanticipated usability issues.

3.3.2.3 Prototyping

We implemented an Android interactive prototype shown in Figure 3.4. We addressed a number of design questions that we wanted to answer through prototyping to inform future design directions. Specifically, we played back four types of Episodes to examine our design. The first type of Episodes was intended for testing normal conditions. The second type of Episodes was to answer pre-selected design questions, including examining the presentation of BusBuddy in particular conditions such as a number of buses arriving at the same place at the

same time, and how ETA of buses were affected by different lengths of "dwelling events"—buses dwelling at the same location for a period of time. The third type of Episodes was those containing atypical movements such as traces with noisy and inaccurate location readings due to a poor GPS signal or with extremely slow movements (possibly due to the recorder forgetting to turn off recording after getting off a bus). Finally, we also played back the Episodes containing randomly selected traces.

We repeatedly played these types of Episodes and observed how the prototype behaved in and reacted to these different conditions. This process had helped us

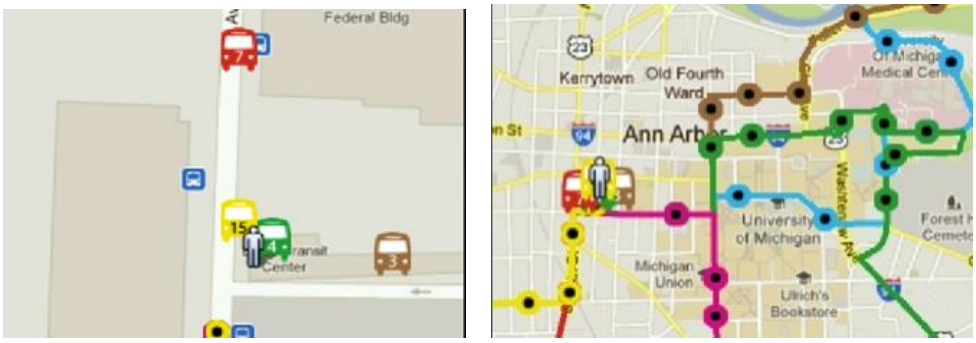


Figure 3.5 We found that it was difficult to distinguish between buses using the same color (left). The issue was more apparent when the map is zoomed out (right).

identify issues that might lead to user frustration. For example, by playing back Episodes containing a number of buses simultaneously stopped nearby the same location, we found that it was difficult to distinguish among bus routes. This made us decide to color-code bus routes and add a route number to each bus to make buses more recognizable (as shown in Figure 3.5). We also observed bus icons overlapping as buses moved closer when the map was zoomed out. This made us decide to adjust the size of the icon or change the presentation of buses based on the zoom level.

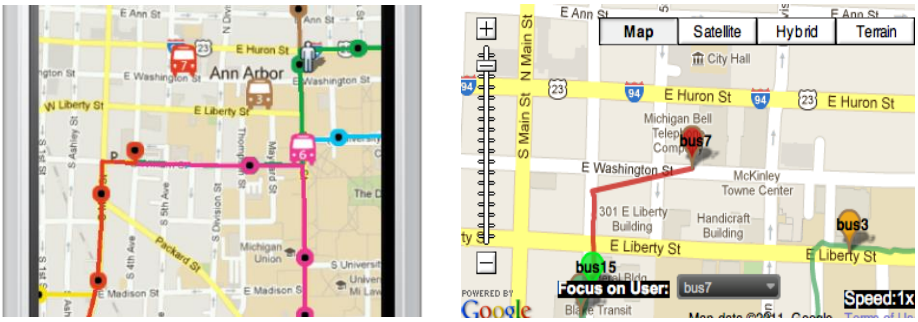


Figure 3.6 Inaccurate GPS locations make bus icons totally off a route (left). From the WorldState window of RePlay designers can clearly see what happened to those traces.

We found several design issue when we played Episodes containing atypical trajectories. For example, we played back an Episode including a bus trace (the Red bus icon representing Route #7 in Figure 3.6) containing jumpy location records (possibly due to a poor GPS signal) and found that the Route 7 bus icon was displayed entirely off the designated path. This not only made us realize that it was necessary to correct or to highlight suspected erroneous bus locations, but also helped us verify that our design of displaying a default bus route on the map was helpful for recognizing whether a bus was off a designated route. Finally, playing back bus traces moving slowly as walking made us consider including a transportation mode detector into BusBuddy and placing a marker next to a bus icon to indicate unexpected movement of the bus.

3.3.2.4 Evaluation

After the first round of reflective prototyping activity in which we had identified several design issues and thus improved the prototype, we were interested in observing how users interacted with BusBuddy. We recruited seven participants who had bus-riding experience to observe how they used BusBuddy to choose which bus to take, and what problems arose to hinder their success. We designed

three tasks in the evaluation study based on our currently available bus location traces.

In the first task, we asked participants to locate a building on the map in BusBuddy and choose a bus route that could take them to the building. In the second task, we told participants that they were at the transit center in the city and they needed to identify a bus route taking them to the specified destination. Finally, in the third task we asked participants to take a route to go to the transit center. The bus stops for each route were at different distances from participants' current location in the study. One constraint we posed was that participants needed to consider which route would take them to the transit center in time in order to catch up a next specified bus route. As a result, participants not only needed to pay attention to the ETA of each route to the bus stop closest to their location, but also needed to consider how far each bus stop was, and how soon each bus would arrive at the transit center.

During each study session, we played back designated Episode(s) with BusBuddy. We played back one Episode for the first and the third task, and two Episodes for the second task. It is noteworthy that in our second task (taking a bus from the transit center to a destination), we intended to let participants see two routes heading to the transit center and then departing for the destination after they chose one of the two routes. However, we did not have a bus trace contained both an arrival and a departure events at the transit center. As a result, we used RePlay to compose this scenario on the fly: playing one Episode in which two bus routes arrived at the transit center, and playing another Episode containing two routes departed from the center.

Through the user study, we were able to witness the helpfulness of displaying bus location and ETA for participants to make decisions regarding which route to

take. In addition, the improvements on color-coding bus routes and adding route numbers next to a bus icon let participants able to distinguish between routes close to each other at the transit center. This improvement made us able to receive comments more regarding challenges with bus riding in general (e.g. being worried about missing a bus when there was a time constraint), and how to make BusBuddy more useful (e.g. showing the direction a bus is heading to) rather than basic usability issues of BusBuddy. We attributed this to early discovering these usability issues through testing BusBuddy with captured bus data. In addition, using the Wizard of Oz approach to playing back captured bus traces during the studies made participants feel bus movements on BusBuddy realistic and close to what they would expect to see if there were using such an application in their daily lives.

However, it should be note that we encountered challenges in creating tasks due to the lack of captured data. That is, we found that we did not possess enough bus traces to create Episodes matching the tasks we originally designed. The gap in the data was because most of our bus traces were relatively short or were only of specific routes, and the team members who designed the tasks were not aware of what bus data we had possessed. As a result, for the tasks we found it easier to revise the task by changing the bus route requirement in accordance with the bus routes that we had collected, we simply changed the task. There was a task that required a number of specific routes with a minimum length in order to generate an Episode that would work for the task, but we were not able to simply change the route requirement. Because the task designer considered it costly to take those bus routes, he decided to drive to simulate those bus routes in order to meet the data requirement of the tasks.

3.3.3 Lesson Learned from the Case Studies

3.3.3.1 Benefits and Limitations of Capture-and-Playback in Prototyping and Evaluation

In our two design projects, we both played back bus location traces in reflective prototyping and system evaluation. In prototyping the two applications, we confirmed that a C&P approach was helpful for validating design, observing how the system responded to specific contextual conditions, and identifying design issues that might otherwise occur when being deployed in the field. For instance, in the LoungeBoard project, it helped us realize that the initial design of border display style needed to be improved in order to better present frequent enter and exist events, and that the collage display style did not work well when the number of occupants was larger than four. It also helped us improve the presentation of bus arrival status. In the BusBuddy project, we played back different sets of Episodes for different purposes, such as testing the design with Episodes representing normal conditions, unanticipated conditions, and particular conditions created for answering specific design questions. This had helped us, for example, improve color-coding of bus icons, identify overlapping bus icons when the map was zoomed out, and recognize a need to show default bus route and adjust bus location when receiving inaccurate location update. Nevertheless, one limitation we found was that the creation of Episodes for prototyping was greatly constrained by the data we possessed—we were not able to examine the prototypes with conditions of interest that we did not have data.

In conducting evaluation for the two design projects, playback was particularly helpful for employing the Wizard of Oz technique. On one hand, we could control the behavior of the prototype according to participants' behavior and to the situation of each study session; on the other hand, we could simulate behavior traces of external entities of interest, such as location traces of buses for both

projects. In addition, playing captured traces made participants feel that the application's behavior was realistic and reflecting what they would expect to see if they were using the application in the real life. However, while playing captured data was for creating realistic experience for participants, the experience we hoped to create was nevertheless constrained by data we possessed. In addition, when using the Wizard of Oz method during the study, we needed to be highly conscious of the situation of each study session in order to avoid any inconsistency between the interface and reality perceived by the study participants. For example, when playing back bus data in the field, it was necessary to pay attention to the presence of real buses nearby to avoid a situation where a bus saw in the field was not displayed on the interface of BusBuddy. To avoid this situation, we brought participants to locations where they would not see the buses.

3.3.3.2 A Capture-and-Playback Plan Evolves in the Iterative Design Process

In the two design projects, both of our capture and playback plan evolved throughout the design process. In particular, we learned that the notion of "what data needed to be captured" was likely to evolve throughout the process. For example, data needed for creating a scenario for a usability test or for a user enactment study would be different from the data needed for reflective prototyping aimed to examine design alternatives or to answer specific design questions. In addition, when we needed a specific set of data traces to create particular contextual conditions, we tended to use a top-down approach to create Episodes. In contrast, when we wanted to test a prototype with unanticipated conditions, we used a more bottom-up and exploratory approach to create Episodes.

In addition to a different strategy to organize and select traces, we also experienced changes in the data capture strategy and goals because of the different needs of data. In the early stage of the process, we tended to adopt an opportunistic data capture strategy to capture as much data as we can. This was because in the early stage of the design projects it had not been clear what kind of behavioral data and scenarios would be relevant and needed for prototyping and evaluation. As we advanced to the later stages where design questions started to emerge, we turned to need data representing specific behaviors or allowing us to synthesize specific scenarios to answer certain design questions and testing certain features of the prototype. As a result, we switched to a more focused and specific data capture while we found the lack of such data. Thus, in later stages it seemed that our data capture was need-driven, toward a "playback-and-then-capture" model. However, we also found it not always feasible or pragmatic to capture data representing specific behaviors when the cost of capture those was high (e.g. needing to spend long time performing the targeted behavior).

3.3.3.3 Annotations Facilitate Organizing, Using, and Communicating Data

Another important lesson we learned is that, in both projects, annotations played a very crucial role in organizing, using, and communicating captured data. One task we found tedious and laborious was reviewing traces lacking useful annotations describing the content of the traces. Because the design team divided data capture and organization to different people (team members and recruited helpers), the team members who later attempted to create Episodes often found themselves lacking the knowledge about which region of a trace was relevant and useful for prototyping and evaluation, respectively. Since a trace could be up to an hour without any useful region to use, we found it quite inefficient to review traces without any annotations describing, "what happened" in those traces. We attributed this issue to the fact that we were not aware of the value of these

explanatory annotations and thus did not instruct the data collectors to be conscious of and to annotate about "events" happening during recording. On the other hand, in the early stage of the design project, it was also unclear what events were relevant and should be paid attention to and noted during the capture. Moreover, it also seems unrealistic to expect data collectors to note any single details of data they capture because of the substantial burden.

To ameliorate this issue, we asked the design team members to review the dynamic of the traces and then add annotations to the file name of a trace file after reviewing each trace. These "post hoc" annotations helped the reuse and organization of the captured data significantly efficient. However, we also found it challenging to use existing tools to organize annotations and map annotations about local events (e.g. a traffic jam) to their corresponding region of the trace besides putting timestamp(s) specifying its start and end point (e.g. traffic jam: 05:30 – 06:45). Thus, it seems to us that a tool visualizing and projecting local annotations to their corresponding segments will considerably facilitate the trace selection process.

Finally, annotations were also helpful for communication among our team. Our team members frequently communicated about extracting portions from existing traces representing particular behaviors; discussed strategies of synthesizing, permuting, and selecting Episodes for different purposes; and shared experiences in and observations on particular traces they had played using RePlay. In these conversations, annotations served as a reference and the common ground among the team. We found that while the team had spoken of data with annotations more, the team not only had a better shared understanding and knowledge of the data, but also knew what annotations would be useful to add during data capture and organization.

3.3.4 Summary

To summarize, the lessons we learned from the two case studies seem to confirm that a C&P approach can support prototyping and evaluating location-aware systems, at least, in helping testing design alternatives, identifying design issues, and answering design questions, and supporting the use of Wizard of Oz and creating realistic experience during a user enactment study and a usability testing. In addition, our lessons learned suggest an iterative and a dynamic nature of capture-and-playback in the design and development process, mainly because a dynamic tension between what data the design team possessed and what data the team needed in different phases of the design process. This may suggest a design space for a mechanism or a platform to request data capture tasks with different instructions (guided, scripted, or opportunistic). Finally, our lessons learned suggest that annotations are useful for using, organizing, and communicating data. We think tools visualizing or presenting annotations of local segments of interest are useful for reducing the need to review an entire trace.

3.4 The RePlay User Study

The goal of the RePlay study is threefold. First, we seek to understand how developers would interact with captured data while working on concrete tasks. Second, we are interested in observing how the features of RePlay helped or hindered developer' development tasks. Finally, we want to learn what additional features a C&P tool should have to better support developers in testing context-aware systems. To achieve these goals, we invited ten developers to use RePlay as they sought to modify a location-aware smartphone application called Here-and-Now (H&N) we had built.

We designed the study to involve tasks that would be feasible to complete within the constraints of a two-hour lab study but that would represent tasks that developers of location-aware systems would need to do as part of a realistic

development process. After a series of pilot studies, we determined that asking participants to write code from scratch was not feasible given the time constraints and the perceived necessity of understanding a significant portion of the H&N code before implementing even a small, self-contained method. Thus, we created tasks that required participants to tune parameters on a set of algorithms that we provided in order to attain acceptable performance. We specifically chose features for which we were certain that there was no perfect algorithm and identifying locally optimal parameter settings would require a certain amount of human judgment. Requiring human judgment, we believed, would necessitate greater interaction with the data, thereby increasing the value we could extract from each session.

3.4.1 The Testbed: Here & Now (H&N)

To provide a testbed for our study, we built a sample application called Here&Now (H&N). H&N, shown in Figure 3.7, is an Android-based smartphone application that allows merchants to offer location- and time-based promotions to mobile customers. In this section, we describe the process we followed to produce H&N and outline the features of the system as it was presented to participants in the study.

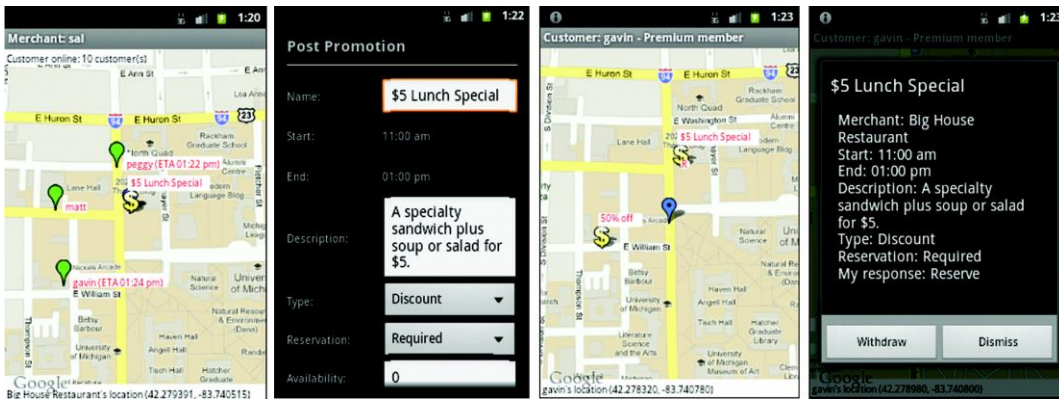


Figure 3.7 Here & Now allows merchants to post (middle left), edit, and delete promotions and view customer locations on a map (leftmost). Customers can view available promotions (middle right) and cancel reservations (rightmost).

3.4.1.1 Initial Goals

Our primary goal in creating an application was to present participants with a plausible application concept that was rich enough to support a wide range of possible design directions. We intended to produce an application that was conceptually coherent but whose design and development was incomplete, so that we could invite participants to help us complete certain aspects of the system. It is important to note that it was not our primary goal to produce a system design that we could defend as commercially viable, demonstrably useful, or free of usability flaws. Indeed, for our purposes it was somewhat beneficial if the system included some design flaws, especially if those flaws are not immediately apparent but emerge upon deeper engagement with the system. Based on the interactions we had with designers and developers during the studies we feel that we were successful in creating a system design that was regarded as plausible based on a high-level description but revealed flaws at multiple levels upon deeper engagement.

3.4.1.2 Application Concept

In choosing a concept for our design, we sought concepts that were similar but not identical to applications with which practitioners and students are likely to be familiar. After brainstorming and developing a number of possible concepts across several weeks, we settled upon "Here & Now." The high-level concept behind H&N is that allows business owners (merchants) to create and post promotions that are distributed to members of the H&N community (customers) via their mobile devices. The H&N concept borrows aspects from a number of commercial applications that were extremely popular at the time of this project, including FourSquare[10], which features location-based promotions, and Groupon[11], which features time-limited promotions. The popularity of these services also meant that participants would probably have some prior familiarity and experience that could make it easier for them to provide feedback on the features of H&N without having participated in earlier project activities.

3.4.1.3 Design and Data Collection

A key part of our early design process for H&N involved the creation of scenarios and storyboards. These artifacts served a dual purpose. While they helped to guide the interaction design and technical requirements for H&N they also helped to guide the collection of sensor data its organization into Episodes. We sought to create a set of scenarios that covered a wide range of possible situations that H&N might need to handle, including different numbers of merchants, different numbers of customers, different types of promotions, different user behaviors (e.g., commuting, wandering aimlessly, going out with friends), and different

---

[10] http://foursquare.com

[11] http://groupon.com

geographical distributions of both customers and merchants. Based on these scenarios, we worked out a data collection plan to try to assemble as many different types of Clips representing as many different situations as possible. Over a period of several weeks, different team members used a combination of RePlay's GPS Capture Probes and commercially available GPS loggers such as MyTracks[12] for Android to collect location traces of their own movements in and around our city's downtown area. The traces represented both natural movements such as commutes, social outings, and errand-running and intentional journeys that served the needs of the collection activity (e.g., walking to specific destination just so we would have an example of that journey in our data set). After filtering and organizing the resulting traces, we had 70 Clips that we organized into nine Episodes representing different situations H&N would need to support based on our own evolving ideas about the application design. All Clips and Episodes were given names and short textual descriptions that were based on the scenarios they represented and their role in those scenarios, such as "Having lunch at the Big House Restaurant" and "Leaving from downtown." In addition, we populated the H&N database with a number of records representing local merchants and fictitious customers with different profile settings.

3.4.1.4 "Final" Design and Implementation

Our final implementation allowed merchants to post, edit, and delete promotions; view customer locations on a map; and configure various aspects of their map view (for example showing and hiding detailed information about customers such as their "membership level"). For customers, the capabilities included viewing promotions on a map; retrieving detailed information about promotions; making a

[12] https://en.wikipedia.org/wiki/MyTracks/

reservation with a merchant if the promotion allows it; configuring their map views (e.g., viewing more or less information about each promotion); and configuring their location sharing preferences (e.g., no sharing, location only, location and name). A number of "advanced" features were also developed for use in the studies but only three were ultimately used, as described later. We designed and built H&N across 8 weeks as an Android application comprised of approximately 7000 lines of code and 40 classes, using RePlay along with the captured Episodes to test and refine the application throughout the period of its development.

3.4.2 Participants

We recruited participants by email and word of mouth who had Java programming experience with a preference for participants who had developed for Android. Six of the people we recruited participated in three sessions as pairs (P1+P2, P3+P4, and P5+P6) and four participated as individuals (P7-P10). All participants had multiple years of programming experience and had written at least one Android program, but only P1 and P10 had more than a year of Android development experience. Participants used the Eclipse 3.3 IDE with the Android SDK (Platform 3.2) installed, as shown in Figure 3.8. All participants were current students at our university (nine graduate students, one undergraduate student), and four of them had prior experience as professional software developers.

3.4.3 Study Tasks and Procedure

After receiving an introduction and signing an informed consent form, each two-hour session started with a demonstration of H&N, RePlay, and Dalvik Debug Monitor Server (DDMS), a built-in Android debugging tool. Participants also received a walkthrough of the critical portions of the H&N code. After the demo, we asked participants to improve one aspect of the application in each of the two tasks shown below. First, we asked them to improve the Estimated Time of Arrival (ETA) calculation. Second, we asked participants to improve an Arrival Detection (AD) algorithm that determined when a customer had "arrived" at a merchant's establishment. For the assigned tasks, our intent was to provide working code that was simple to understand, but that had detectable flaws when

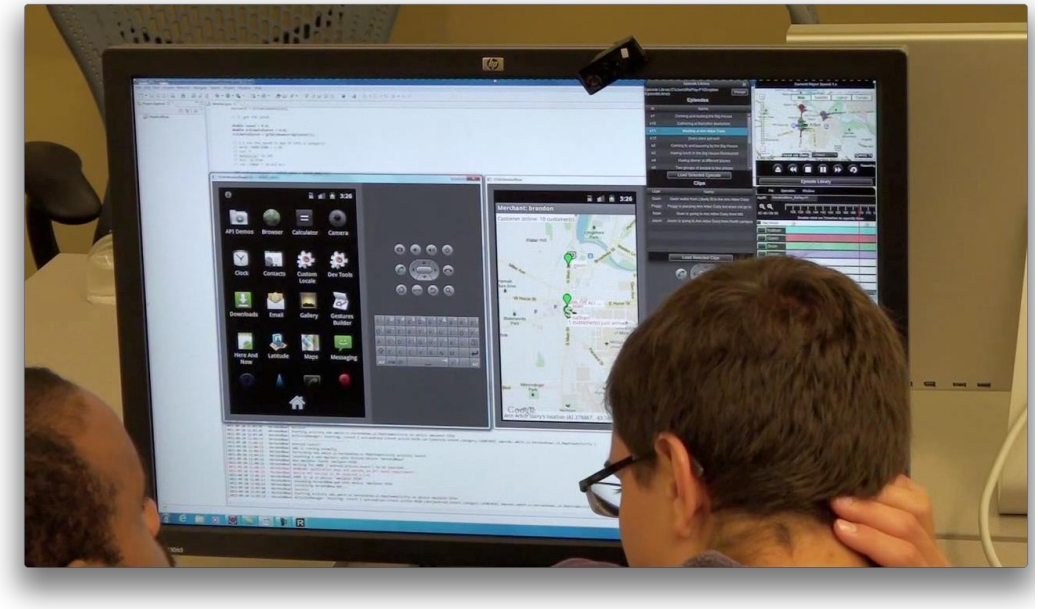


Figure 3.8 We provided participants with the H&N code as well as RePlay and DDMS for playing the captured data to perform the ETA and Arrival Detection tasks

tested with real data. Below we provide more details of the two tasks:

Task 1: Select and tune an Estimated Time of Arrival algorithm. Participants were presented with three different implementations of computing a customer's ETA. All of them depend on knowing the customer's distance from the destination and computes the ETA by computing the customer's speed. The first assumes a constant speed for all customers, regardless of their actual speed. The second computes their average speed over a short window of time. The third approach uses the windowed average to infer the customer's mode of transportation (walking, driving, cycling, riding the bus) and then chooses a constant speed that is appropriate for that mode. Participants had to experiment with each approach and decide which implementation represented the best approach. Once they chose, they were to tweak parameters or write new code for that algorithm that will give the most useful information to the merchant in the broadest set of cases.

Task 2: Tune the Arrival Detection (AD) algorithm. H&N includes a method to detect when a customer has arrived at a destination. The method has two parameters: a range parameter that determines how far away from the target a customer can be and still be considered "present," and a time parameter that determines how long a customer must be within the specified range before being marked as "arrived."

For both tasks, we were interested in seeing not just whether participants could "solve" the tasks (which was unlikely given that both ETA and AD are quite challenging to solve in the general case—see (Marmasse & Schmandt, 2002)), for example), but whether and how they could use the provided data to better understand the problem and identify issues that would need to be addressed for a more robust solution.

We provided participants with the H&N code as well as two tools for playing the captured data: RePlay and DDMS. DDMS comes with the Android SDK and

includes the ability to manually set the location of an attached Android device or emulator as well as the ability to feed location data from a GPS Exchange (GPX) file containing a time-stamped location trace. All captured traces were provided in GPX format as well as made accessible through the RePlay Episode Library. Participants used both DDMS and RePlay to perform both tasks, and the presentation order for DDMS and RePlay was varied both within and across sessions. Task order was not varied.

We encouraged participants to think aloud, and attended carefully to the choices they made about which data to use and how those data were incorporated into their programming and testing activities. At the end of the session, we conducted a short debriefing interview to know about their experience using both RePlay and DDMS and to obtain their feedback on both tools. Participants received $40 (increased from $30 after low initial response to recruiting attempts) for their participation in a two-hour session.

All sessions were video-recorded and interactions with RePlay were written to a log. Recordings and logs were reviewed to examine the rationale behind every interaction with RePlay in the context of the participant's overall progress on the tasks. From this analysis, we were able to identify themes relating to how participants tried to make sense of the data, selected data relevant to each task, sought to correct deficiencies in the available data, and used data in the course of modifying the code and thinking through possible solutions.

3.4.4 Study Results and Findings

Participants clearly had trouble doing their tasks using this version of RePlay; most participants struggled to make progress. Only one participant (P7) was able to modify both the ETA and Activity Detection (AD) algorithms and provide a coherent explanation for why his changes represented an improvement. Of the

others, only P9 tested all three ETA methods during the course of the session and was able to verbally compare their performance despite not making any changes to the code. All others tried only one or two ETA methods during the allotted time. None of the participants beside P7 were able to get the AD code to detect an arrival—mainly due to not being able to find a suitable Clip—even though two others tried to adjust the AD algorithm parameters to be more permissive.

Despite not being able to make improvements to the code, most participants made verbal statements indicating that they had a better idea about how to approach the problem after working on the task. Some of the specific insights came directly out of interacting with the data. For example, P5+P6 changed the AD distance threshold to 5 meters and the timeout to 180 seconds based on their discussion. They recognized that these parameters were unworkable, though, when they tried it with actual data—noting that noisy data would regularly report a user "jumping" more than 5 meters when they were clearly in one place, and seeing how long it "felt" to wait 3 minutes before an arrival was reported. They instead changed the parameters to 10 meters and 30 seconds, which they felt made more sense based on the data. Likewise, P3+P4 noted that relying on the distance from the customer to the merchant was not a viable approach on its own, because many of the traces did not head directly towards the merchant. After watching a particular trace with a number of direction changes, they worried about a customer "walking backward" and noted, "if the user walks around, this algorithm will explode."

Though participants eventual came to an improved understanding of each assigned problem based on the data, it was clear that finding and using traces was not easy, and slowed participants' progress on the tasks. We observed three different aspects of interacting with the data that presented challenges. We believe

these issues are not specific to RePlay, but generalize to problems that developers would have interacting with captured data more broadly. They are:

3.4.4.1 Selecting Examples

Participants spent a good deal of time trying to find traces that could be used for testing the relevant H&N features. The process of finding traces included both sensemaking (i.e., understanding the nature and scope of the data and trying to match it to the task) and targeted search. For example, while sometimes participants looked for traces with specific characteristics, participants often attempted to find traces on the basis of whether or not they contained a particular "critical section." In both the ETA and the Arrival Detection task, for example, participants specifically wanted to find traces where it was easy to identify an "arrival"—i.e., the precise point at which the user in the trace arrives at a particular destination.

Finding such an example was not always easy and efficient. To find good examples, participants made frequent use of RePlay's facilities for providing information about traces. For instance, all of the participants found that Preview was a useful feature for identifying candidate traces. In some cases, the descriptions provided information about the original capture situation that was helpful. For the ETA task, it was useful to know what transportation mode(s) the user had employed since this information was used in one of the algorithms they needed to evaluate. Some trace descriptions contained such information, which helped participants to locate them easily. However, participants did not always find these facilities adequate for identifying traces they wanted to use for more extensive testing. Rather, many participants engaged in several rounds of loading traces into the Player and watching the user traces play out in the World State window and/or H&N application. They sometimes even played through many

traces, often in their entirety, in order to determine whether they were suitable for the task. Part of the problem was the lack of appropriate annotations, but part of it was that the attributes of interest were dynamic in nature (e.g., "stops" and "speed"), and, given the limitations of RePlay could only be observed by watching the trace play out. In several cases, the difficulty of search resulted in participants using a trace that was not well suited for the task. In the ETA task, for example, P5+P6 used a particularly noisy trace for most of the task because it was the first one they found that arrived at their chosen destination. However, once such a good example was eventually found—most participants would then stick with those traces for the duration of their work on the given task (typically 10-20 minutes).

3.4.4.2 Modifying Data:

Participants in five of the seven sessions used at least one transform. P08 and P10 all used the Signal Lost Transform to see how the various ETA algorithms would respond to a user's lost GPS signal. These participants reported that their prior experience with mobile app development led them to worrying about how the algorithm would respond to this common situation. P08, upon observing the behavior of all three ETA algorithms under the signal lost condition concluded that none would be adequate for a real deployment and each suggested that a new, more sophisticated algorithm would need to be developed. P05/P06 used the Freeze Transform to simulate an arrival at a destination in order to more clearly understand how the Arrival Detection algorithm was working. P01/P02, P07, and P10 used the Identity Transform to quickly see how the system responded differently to users with different characteristics (in this case, users who do and do not have a reservation with a particular merchant).

However, in some cases, participants felt it would be more straightforward to just make the trace they wanted. For example, P7 at one point applied a transform to a trace to simulate an arrival event, but he struggled to configure the transform to show up in the right part of the trace. P1 noted that it would be easier for him to just create what he was looking for. *"If I could just draw a path with a customer going directly to it, and a path with a customer going right past it, I would be able to see which one works."* He later clarified, *"I work from a simple case first, and then I make more complicated cases...."*

The desire to test with both simple and problematic examples was reflected in a number of other sessions, where participants either discovered or anticipated potential problems in the course of looking for typical cases. For example, P7 specifically chose a trace with a significant period of lost signal for testing. P8 and P10 noticed signal-loss in other traces during the exploration phase and used the signal loss transform to create such a period in later traces.

3.4.4.3 Controlling playback during iterative testing:

Additional challenges arose when trying to use a trace for iteratively testing and improving the tasks' algorithms. Most notably, participants generally chose traces based on a particular subregion—be that an arrival event, a period representing particular movement characteristics, or a region of lost signal, usually the five or ten seconds representing a person's arrival or a one minute period in which the three ETAs exhibited dramatic changes. This small bit of data, however, could be used extensively. In Session 3, the participant(s) used RePlay to play a particular 35-second long subsection of the selected clip fifteen times during the seven minutes (with double speed) they were working on the ETA task. However, when repeatedly playing back a trace, it was cumbersome to re-find the region of interest. For some uses, the ability to repeat a particular subregion of the currently

loaded Episode was helpful, but this was inadequate when there were multiple events of interest within a trace or a set of traces.

In addition, monitoring the progress of a trace during testing was challenging. Several participants were interested in seeing how the occurrence of the event in question triggered changes in the H&N user interface, and it was difficult to monitor the various aspects of the RePlay UI at the same time. P7 noted that it would be convenient to be able to see more in the World State window about what was happening in the trace, such as directly indicating when an arrival event had occurred.

3.4.4.4 Summary

The programming study looked at how developers of location aware systems could use captured data for testing and improving features. While multiple tools exist for capturing and rendering such data (including the Android SDK's own DDMS tool), the ability to organize, preview, select, transform, and control the playback for dozens of traces was seen as the primary advantage of RePlay over DDMS that help developers find and use captured data more effectively. These capabilities, along with captured data, was useful for helping our study participants refine their understanding of how to approach ETA and Arrival Detection for H&N. However, we identified a set of challenges that held most participants back from successfully implementing changes within the available time. In looking at the shortcomings of RePlay's facilities for finding, modifying, and controlling traces, we were able to see opportunities for offering better tools to more effectively support managing and using data in context-aware development.

3.5 Initial Improvements: TraceViz

Based in part on the results of the user study, we set out to improve the tools available to developers for working with contextual data. The study indicated that the ability to access captured data would be helpful, but that additional support would be needed for finding, modifying, and playing back such data. Our first effort in this arena was to develop TraceViz, a visualization tool for browsing and selecting GPS trace data (Y. Chang et al., 2012). TraceViz specifically aims to address the difficulty of finding examples from a large number of location traces. That is, when an increasing amount of GPS location data has been captured and aggregated, it is difficult to select particular location traces for testing. While general-purpose geovisualization tools like Google Earth Desktop[13] can be used to visualize location traces on a map, it remains challenging to explore, filter, and select individual location traces when presented with a large set of data.

To address this problem, we developed TraceViz to allow location-aware application designers and developers to filter by *brushing* to directly indicate geographical regions and trajectories of interest. TraceViz provides three brush modes—*Reselect*, *Intersect*, and *Union*—to allow designers to flexibly narrow down or expand filter criteria. When a brush stroke is drawn, the system calculates the similarity score of nearby traces and adjusts their visual saliency: only highly similar traces remain salient on the map to make it easier to highlight and select those relevant traces. After selecting traces of interest, the designer can import the selected traces from into their chosen playback tool.

TraceViz is the first tool to leverage *brushing* to help explore, filter, and select location traces for testing location-aware applications. By making it easier to find

[13] http://www.google.com/earth/explore/products/desktop.html

and select relevant traces, TraceViz encourages location-aware application designers and developers to validate the design of their location-aware applications with a greater variety of location traces, in turn producing higher-quality systems.

Here, I describe a scenario in which a location-aware application designer would like to use TraceViz to find particular location traces.

3.5.1 A Scenario of Using TraceViz

David is an LBS designer who is developing an application that recommends promotions to a user based on their current location and recent trajectory. Knowing that such an application would need to intensively respond to location updates at various places in the downtown area, David's team recruits prospective users to collect a large collection of location traces for testing. As the application prototype evolves, David wants to test the prototype with a number of test cases that involve a traveler passing by specific promotions. To find traces that travel specific routes will require David to review all of the traces his team has collected, so he turns to TraceViz.

David launches TraceViz and uses the search box to center the map on a certain restaurant. He enables the brush mode and brushes a route along a street on which a promotion is located. This results in only five traces that pass through the area he brushed on the map. David selects one location trace from this set, and uses it for testing the application.

3.5.2 The TraceViz Interface

TraceViz consists of three major components, as shown in Figure 3.9. The TraceViewer (Figure 3.9b) visualizes location traces and allows users to brush to filter traces. Traces are color-coded based on whether they are highlighted, selected, or brushed. Users can hover over a trace to view detailed information in the TraceInfo Panel (Figure 3.9c), and they can click on the trace to select it. The Control Panel (Figure 3.9a) allows filtering based on trace duration and distance, and additionally provides controls for users to select brush modes, adjust brush thickness, and set a brush tolerance threshold. Users can utilize the time and physical length filters location traces by time duration and the physical length of the traces in addition to brushing. The brush thickness slider allows users to adjust the coverage of a brush stroke. The tolerance threshold slider controls whether a

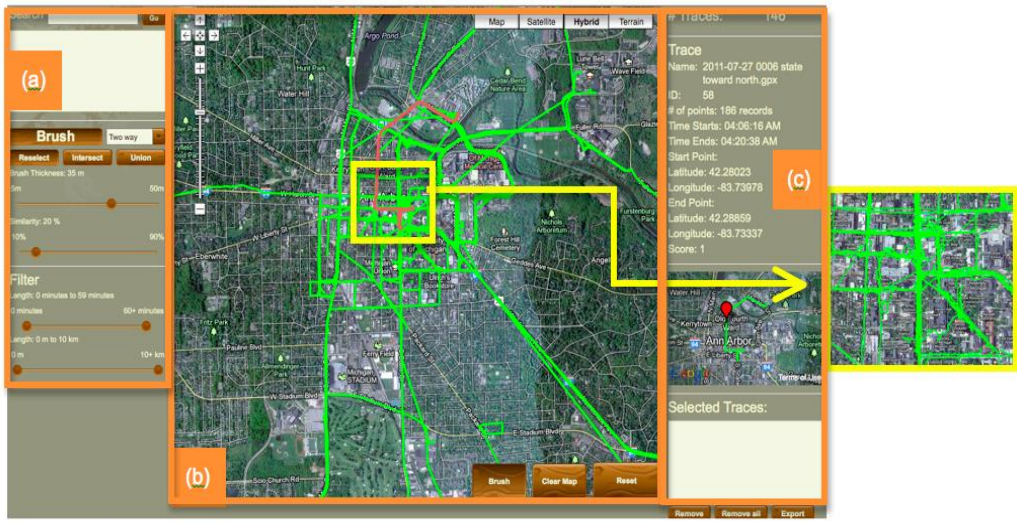


Figure 3.9 (Left) The main interface of TraceViz consists of three components: a Control Panel (a), a TraceViewer (b), and a Trace Info Panel (c). The more the location traces being visualized, the more difficult one can distinguish among location traces (Right).

brushed trace should be displayed based upon its similarity to the brush stroke. The brush mode buttons allow users to switch among the *Reselect*, *Intersect*, and *Union* modes. Once one has selected a set of location traces, the selected traces can be downloaded in a preferred file format (e.g., GPX) or loaded directly into the RePlay system.

TraceViz was built using the Flex 4.1 SDK and uses the Google Map API[14]. It can run on any browser with Flash Player 10 installed. It connects to a MySQL database that contains the traces, and has the ability to import traces in standard file formats such as GPX.

3.5.3 Brushing to Explore and Filter Traces

Brushing is a direct manipulation technique that TraceViz employs to allow users to specify trajectories on a map to filter traces that are "similar" to the brushed stroke. When a brush stroke is drawn on the TraceViewer, we determine which traces are "similar" to the stroke as follows:

1. We identify a set of candidate traces by including all traces that have at least one point within the stroke's candidate area, which we define as a rectangular area around the stroke that is 2 * brush thickness pixels larger than the stroke bounds in all directions. The goal of this step is to reduce the number of traces that are subsequently considered while retaining enough information about each trace to be able to determine the degree to which it is aligned with the brush stroke.

[14] https://developers.google.com/maps/

2. For each candidate trace, we compute the brush-to-trace similarity by computing the proportion of brush points that are near (i.e., within brush thickness of) at least one trace point. This gives a higher score to traces that are well aligned with the brush stroke throughout the stroke's entire length. It also penalizes traces that have few points that fall within the brush stroke, whether due to misalignment or due to sparse data.

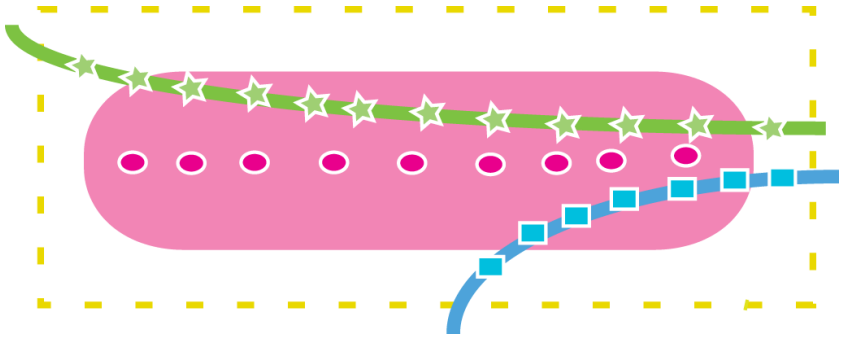
3. We also compute the trace-to-brush similarity for each trace by computing the proportion of trace points that are near at least one brush point. This gives a higher score to traces whose nearby points lie mostly within the brush area and penalizes traces whose trajectories diverge from that of the stroke. While in most cases the brush-to-trace and trace-to-brush scores are redundant, both are needed to deal with cases where the trace and brush point densities differ.
4. Finally, we compute the overall similarity by averaging the trace-to-brush and brush-to-trace similarity scores. Note that all traces that lie outside the candidate area are assigned an overall similarity of zero.

Each trace is then rendered on the map according to its similarity score: first, all traces with similarity scores below the user-defined tolerance threshold are given alpha values of zero, and all other traces are given alpha values proportional to their similarity score.

As an example, consider the situation depicted in Figure 3.10. Trace 1 (green stars) is assigned a brush-to-trace similarity of 1 since all of the brush points are near to at least one trace point. It receives a trace-to-brush similarity of 0.83 since

Figure 3.10 Two candidate traces intersect the candidate area (dashed yellow line). The top trace (green stars) is more similar than the bottom trace (blue squares) because more of its points lie within the brush stroke region (pink oval).

10 of 12 candidate points are near at least one brush point. The overall similarity is therefore 0.92. Trace 2 (blue squares), on the other hand, receives a brush-to-trace similarity of 0.44 since 4 of 9 brush points are near at least one trace point, and a trace-to-brush similarity of 0.71 since 5 of 7 trace points are near a brush point. Trace 2's overall similarity is thus 0.58.



In order to give LBS designers additional control, TraceViz provides three brush modes—*Reselect*, *Intersect*, and *Union*. In *Reselect* mode, every stroke generates a new result. In *Intersect* mode (shown in Figure 3.11), each stroke after the first refines the filter to show only traces that pass through all strokes, whereas in Union mode each stroke widens the filter to include traces that pass through any stroke. As an example, we return to our earlier scenario to illustrate the use of Intersect mode:

David wants to find a trace passing by both a coffee house and a bookstore that are on two different streets. He uses the Intersect mode to brush two strokes near the coffee house and the bookstore, respectively. However, he finds that the filtered traces are still many and overlap one another. As a result, he brushes the third stroke on another street to refine the filtered results. Now David can easily distinguish the traces and select them for testing.

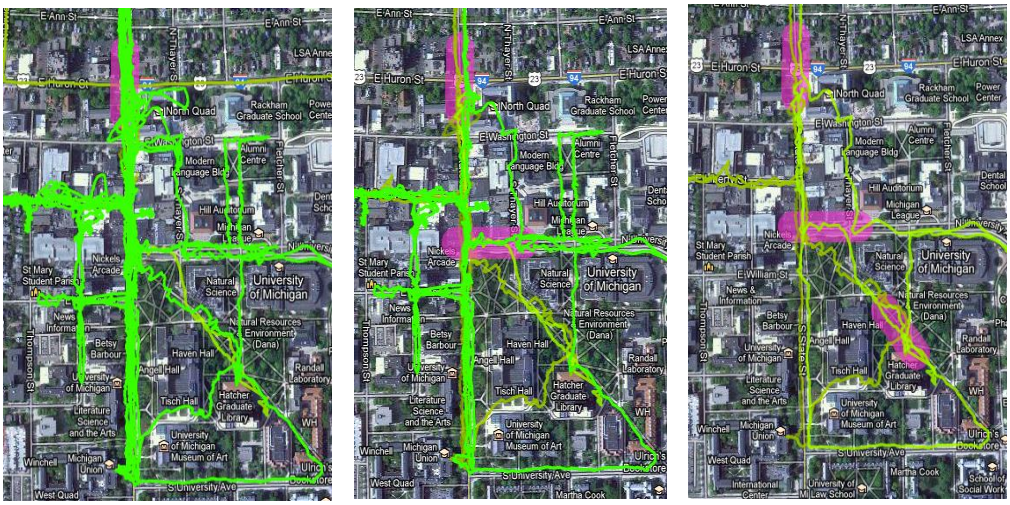


Figure 3.11 The Intersect brush mode allows a location-aware application developer to refine a filter by adding additional brush strokes, as shown here from left to right.

3.5.4 The TraceViz User Study

To observe whether and to what extent TraceViz can help location-aware application designers and developers efficiently select location traces, we conducted a usability evaluation for TraceViz. We recruited eight people with experience in designing or developing mobile applications to participate and asked them to use TraceViz to perform four tasks in which they selected location traces to load into RePlay. Their goal was to test aspects of Here & Now (H&N), the promotion LBS mentioned earlier. Our tasks were all based on finding traces that would be suitable for testing this feature, as shown follows:

1. Task 1: Participants were asked to practice selecting a trace by brushing.
2. Task 2: Participants were asked to find a trace that allowed testing different display ranges (0.5 km, 1km, and 2km) and show that H&N respected users' preferences in all cases.
3. Task 3: Participants were asked to find traces that passed by at least two active promotions.
4. Task 4: Participants were asked to find two traces that approached a promotion from different directions in order to show H&N working with multiple simultaneous users.

We provided each participant with 200 GPS traces collected in Ann Arbor, Michigan, USA (the city where our study took place). All participants received a demonstration of TraceViz, RePlay, and H&N at the beginning of the session. Upon completing the tasks they were asked about their reflections on using TraceViz. Participants were encouraged to solve the tasks in any way they wished, and were not directed to use particular features of TraceViz. Video and audio for all sessions were captured, along with detailed session notes, and these

data were reviewed to assess and interpret task success, critical incidents, and participant satisfaction.

3.5.4.1 Study Result

Seven of the eight participants were able to finish all four tasks with little or no assistance. Most participants were able to find suitable traces within one or two attempts for each task. This suggests that TraceViz is able to support location-aware application S designers' in efficiently finding and selecting traces for testing a location-aware application. Moreover, participants developed several different strategies of using brushing for finding suitable traces. For example, while some participants used the Intersect mode for finding traces passing by two specific areas, other participants used it simply for reducing the number of traces on the map. In addition, four distinct styles of brushing were observed from the participants' sessions, as shown in Figure 3.12.

The first style was to draw a precise stroke alone a route on the map, based on what participants thought a target location trace should pass along. The second style was to draw several points on the map. This brushing strategy was used when participants wanted the location traces that passed through all the drawn points. The third style was to draw a long stroke (Figure 3.12 c) that went through several streets. This was used when the users wanted to find the location trace that go along two specific routes. And the fourth style is to draw an area to capture any location traces that pass through the area. Although we asked

participants why they drew in a specific way, unfortunately, given the small number of participants, we did not find a general pattern regarding when a particular strategy would be used for a certain search.

In general, TraceViz enabled participants to filter and select suitable traces or testing the H&N app. However, we also uncovered several shortcomings of TraceViz that need to be addressed in future tools like TraceViz. Some shortcomings were basic usability problems, such as confusion about the brush

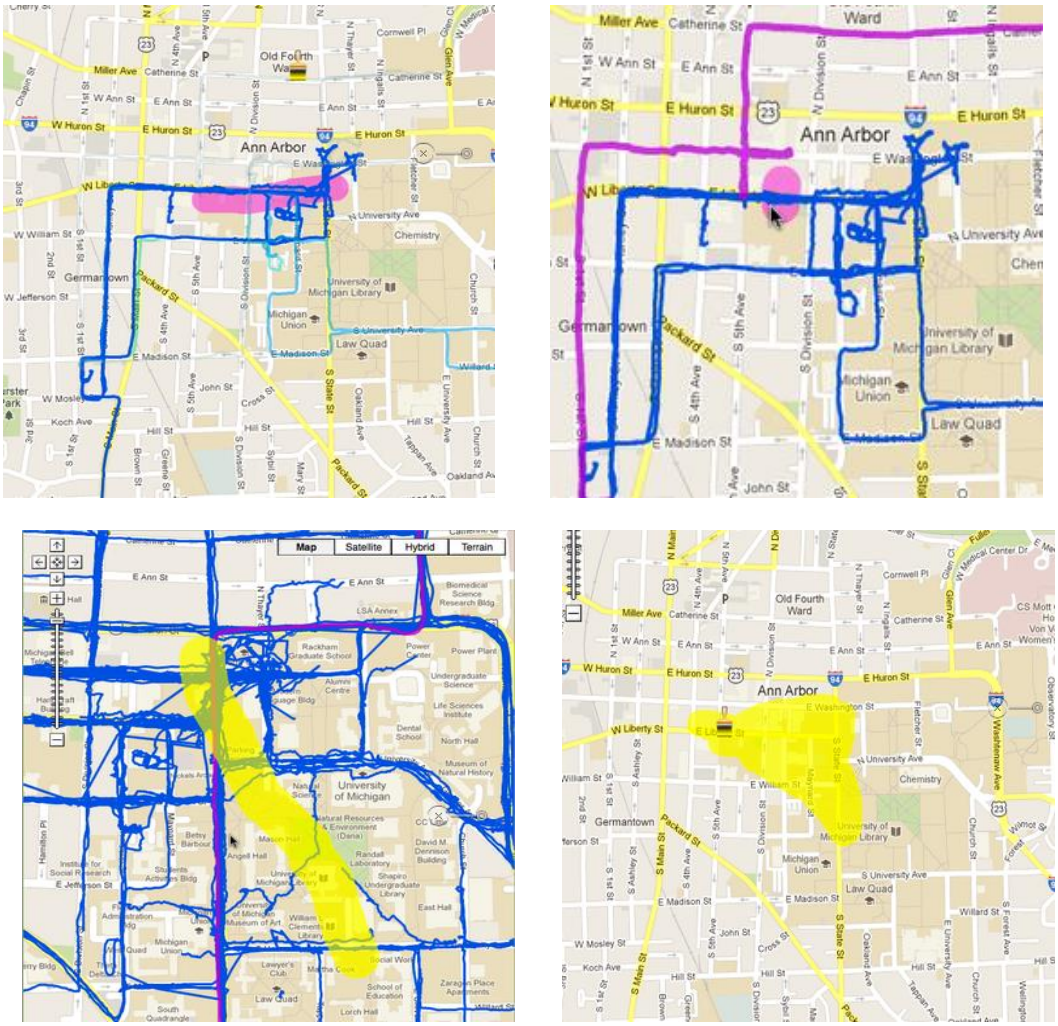


Figure 3.12 Participants had four different brushing strategies. From the leftmost

mode names and difficulties switching between brushing, selecting traces, and panning the map. However, what emerged from the evaluation was that most participants struggled when choosing a trace with unexpected characteristics, including changes in direction and speed or poor signal quality. For instance, P2 selected a trace with sparse location records (probably due to poor GPS signal), and it took her a long time to accomplish one of the tasks as a result.

The issue of lacking information about sensor traces has been noted in the RePlay user study. But in that study, this issue was not as apparent as it was in this study because the participants in the RePlay user study encountered a problem of exploring and selecting data early before they actually used the selected traces. Because TraceViz eases the process of exploring and selecting location traces. Participants were directly exposed the next challenge—dealing with the non-geographical features of location traces that were not obvious on the TraceViz interface.

3.6 Towards A Comprehensive Toolset: Capla

To more fully address the issues identified by the RePlay user study, we set out to build a comprehensive set of tools that would support the selection, modification, and efficient playback of contextual data during the development process. Our resulting toolset, called CaPla (for Capture and Playback), is an integration of RePlay and TraceViz, with several other major technical enhancements. Specifically, CaPla consists of three major components. The Clip Browser is based on TraceViz and contains all of TraceViz's dynamic query and selection capabilities while adding the ability to query and visualize data based on higher-level attributes. The Clip Editor is a new tool that allows developers to modify trace data at multiple levels of granularity. The Clip Player is based on the original RePlay and contains all of its original capabilities with two exceptions: the Transform capability has been moved to the Clip Editor, and the Clip Library has been replaced by the Clip Browser. Additionally, the Clip Player adds greater

control over playback by allowing users to "snap" to semantically meaningful segments of longer traces. RePlay's Capture Probes and Episodes are also included in CaPla essentially unchanged, and are not discussed further in this paper. Compared to RePlay and TraceViz, CaPla adds three major technical enhancements:

- A new tool, the Clip Editor, to support modification of data.
- Augmentations to the Clip Browser and Clip Player to provide visualization and control capabilities based on high-level events contained within traces.
- A common infrastructure to provide uniform mechanisms for visualizing, interacting with, and manipulating data in terms of semantically meaningful events as well as low-level data.

We will now provide an overview of each of CaPla's major components, followed by a description of the common infrastructure that integrates the CaPla tools. At present, our support for GPS is the most mature and provides the best illustration of CaPla's capabilities.

3.6.1 The Clip Browser

As noted, we integrated TraceViz to address developers' need to explore available data and select traces for use in testing. TraceViz was incorporated into CaPla and renamed the Clip Browser (see Figure 3.13) to maintain a consistent naming

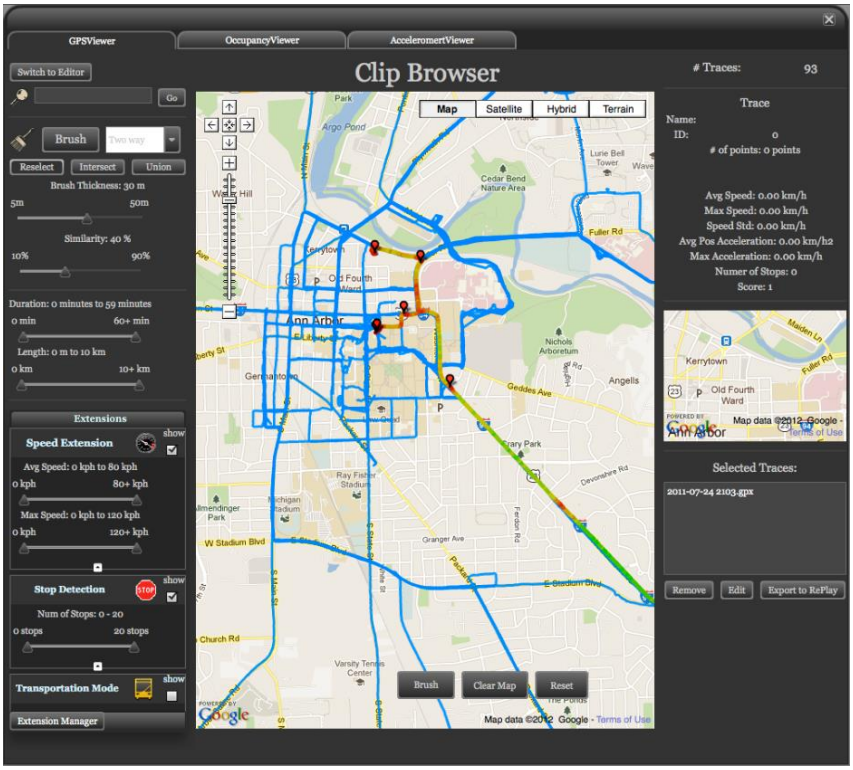


Figure 3.13 The Clip Browser features dynamic query controls and selection brushes for exploring and selecting data examples. Extension-provided Markup is shown for selected Clips, allowing the developer to see particular attributes of the data within the Clip.

scheme. Additionally, the Clip Browser was augmented with the ability to filter based on higher-level events within the data, and to visualize such events within traces. Including these capabilities in a flexible way required implementing a general extension mechanism at the toolset level, which is described below. However, from the user perspective, these additional capabilities are integrated seamlessly into the original TraceViz user interface by adding additional range sliders for specifying dynamic queries and decorating selected traces with information about events and other attributes within the traces.

3.6.2 The Clip Editor

In the Clip Editor (see Figure 3.14) users can add, move, or delete specific points through direct manipulation, This "raw data" capability is standard in other GPS editing tools (e.g., GPX Editor[15]), and is required both for generic data cleaning (e.g., eliminating noise), and also for simulating certain phenomena. For example, the ability to add and move GPS points would have satisfied P1's desire to "draw" particular movement traces. In addition, the Clip Editor allows higher-level events (e.g., stops, changes in speed) to be simulated as well by allowing Transforms to

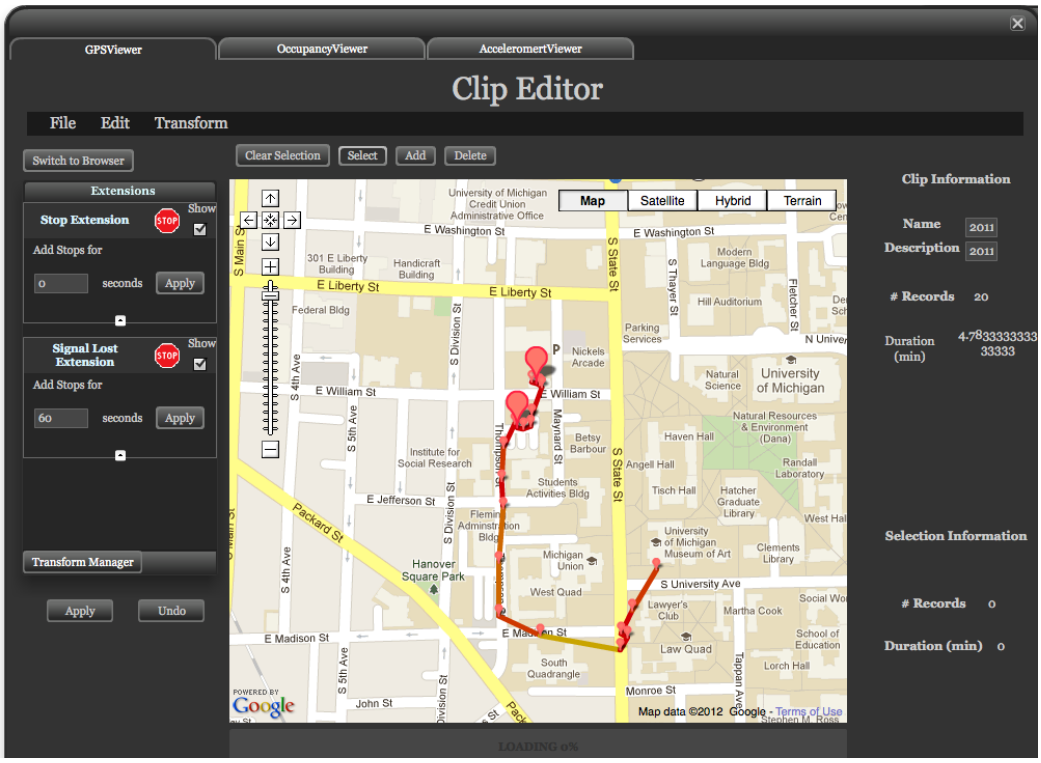


Figure 3.14 The Clip Editor allows developers to manipulate Clip data via direct manipulation or using Transforms, which are Extension-provided operations that manipulate the Clip data at a high level.

[15] http://sourceforge.net/projects/gpxeditor

be applied to selected Tuples or sets of Tuples. As with the filter UI controls in the Clip Browser, the user interface controls for applying and parameterizing a Transform are integrated into the Clip Editor user interface, though the functionality is provided by the same Extension mechanism that is used to enhance the Clip Browser.

3.6.3 The Clip Player

The core functionality from RePlay—the ability to stream previously captured data into a context-aware application under development—is encapsulated into the Clip Player in CaPla. In addition to providing a multi-track timeline view, media playback controls (e.g., play, pause, speed up, slow down), and a "World State" viewer to track the current state of the playback as it progressed, the Clip Player has been enhanced to support streamlined playback based on high-level events

In the RePlay user study, we observed that participants struggled to control playback efficiently, and participants' progress hampered by needing to play through unnecessary portions of data in order to reach the segments of interest. The ability to add Annotations to the timeline of a trace was therefore included in the Clip Player, allowing developers to bookmark time periods of interest within traces to make it easier to return to them later. In addition to supporting manual annotations, the high-level events automatically detected by Extensions can also be made visible in the Player, and can be used to control playback. As shown in Figure 3.15, Annotations are shown on the Player timeline, and a dropdown list allows the developer to "snap" the playhead to the beginning of any Annotation currently loaded into the player. Additionally, to help the developer monitor critical events and changes in the dynamic state during playback, the World State window also displays a message when the annotated data is active on the timeline.

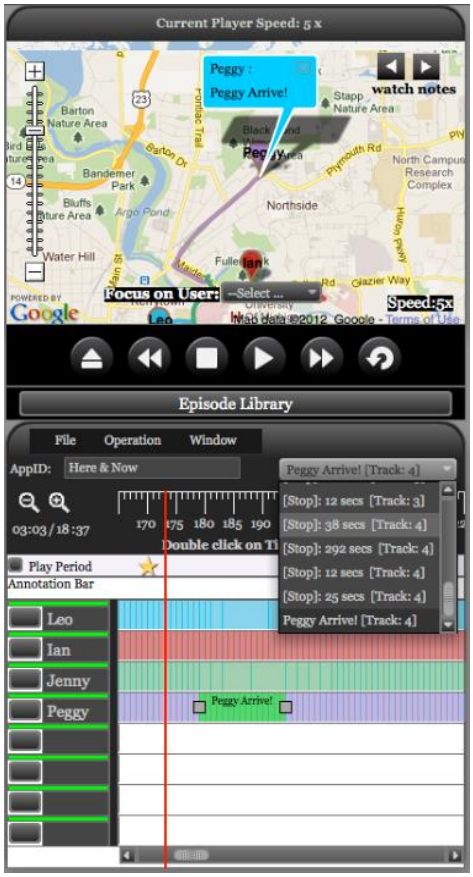


Figure 3.15 Both automatically and manually generated annotations are used to change the current playback state. In addition, annotations are shown in the World State window as well as in the timeline to allow monitoring semantically meaningful events.

3.6.4 Extensions: Labels, Markup, and Transforms

In analyzing the results of the RePlay user study, we realized that, while participants were interested in specific attributes of the data (e.g., stops, noise, signal loss, speed, transportation mode), the attributes of interest would be highly dependent on the nature of the application and even the nature of the particular feature being worked on. Thus, the list of derivable attributes could be quite large and difficult to know in advance. Moreover, since CaPla is intended to serve as a general-purpose toolset for capture and playback of contextual data, it was

important to design the ability to interact with high-level and derived attributes of the data in a way that would generalize to a large range of sensor data. Our solution was to design an Extension mechanism that would make it easy to add and subtract data processing functionality from CaPla tools on an as-needed basis.

At the most basic level, CaPla Extensions operate in a similar way to how extensions work in many other systems: They are bundles of executable code that can be enabled or disabled in accordance with the user's needs and preferences. Extensions in CaPla, however, are specifically intended to enhance the semantic-level aspects of the captured data in ways that more closely match users' task goals. Each Extension is an encapsulation of a particular type of semantic processing and provides four capabilities related to its dedicated purpose. Creating an Extension consists of implementing a set of pre-defined methods, each of which will be invoked by CaPla tools under particular conditions, including `generateLabels(Clip)`, `generateMarkup(Clip)`, `filterClip(Clip)`, and `applyTransform(Tuple[],Clip)`. This allowed us to add the following functionality to CaPla:

1. Indicate regions where semantically meaningful events (might) occur by applying custom labels to regions of the raw Clip data,
2. Render labeled data as visual annotations using markup generated by the Extension that superimposes higher-level information on top of the rendered raw data,
3. Allow users to query the set of available Clips using the Extension-generated labels, and
4. Transform selected data to *create* a region that exhibits semantic properties relevant to a user's data needs.

As an example, consider a "Stops" Extension. This Extension is responsible for giving the CaPla the ability to understand when a region within a GPS trace represents an instance of someone stopping for a period of time. With the Stops Extension installed and enabled, CaPla is able to automatically apply annotations to a set of Clips indicating where possible Stops are, display those as annotations in different tools, provide the ability to query Clips based on attributes specific to Stops, and Transform a Clip to contain a Stop at a particular point for a particular period of time.

Extension functionality is hooked into CaPla in three places: the Labeling and Markup pipeline, the Clip Browser filter controls, and the Clip Editor transform controls. We will describe the Labeling and Markup pipeline first, and subsequently describe the integration with the Clip Browser, Clip Editor, and Clip Player.

3.6.5 The Annotation and Markup Pipeline

When Clips are imported into RePlay they are processed by the Annotation and Markup pipeline. The input to this pipeline is raw Clip data, and the output is Clip data that has been augmented with Labels and Markup. As shown in Figure 3.16, the pipeline first calls `generateLabels(Clip)` on each enabled Extension for each Clip that has been imported. This allows the Extension to apply annotations based on the Clip data using its own internal logic. In this phase, the Extension also builds one or more indices associating the Clip with Extension-specific properties (e.g., number of Stops) that can be used for filtering later. The resulting data structure is an augmented Clip—i.e., a Clip with additional fields representing the Annotations. While the Labels are stored within the Clip data structure, the meanings of the Labels are opaque to the CaPla tools, and must be passed back to the Extension so that CaPla will know how to display them. In the next stage of the pipeline, `generateMarkup(Clip)` is called for each Extension, again for each Clip. In this method, the Extension translates the Labels that it understands

(ignoring all others) into a set of Markup metadata, which will provide hints to the various RePlay renderers. For example, the Markup language for allows an

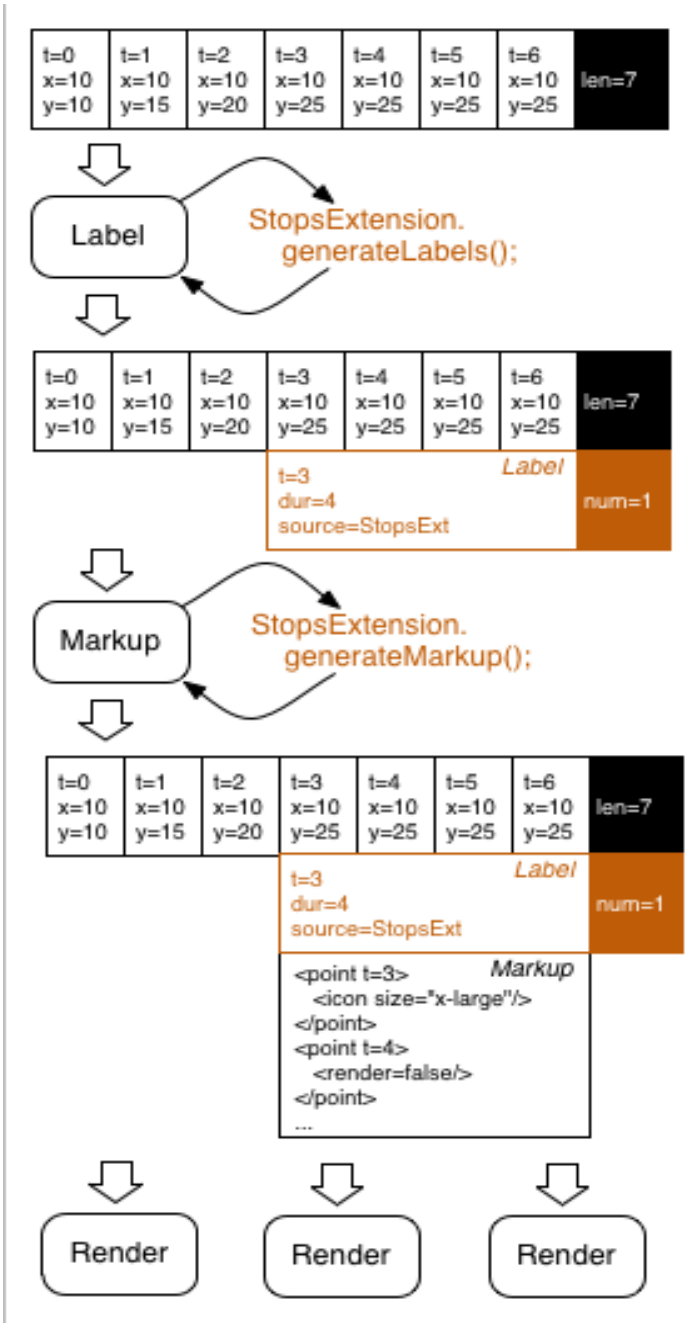


Figure 3.16 A Clip is labeled by the Stops Extension to indicate that a Stop occurs from t=3 through t=6, and the number of Stops is 1, The Clip is then marked up with hints that tell the CaPla renderers how to display the Stop.

Extension to override the default rendering for Points (the Tuples representing individual sensor readings) and Edges (the lines drawn between Points), specifying properties of these elements such as color, size, line thickness, and icon.

There are four different components within the current CaPla toolset that render Clip data: the Clip Browser, the Clip Editor, the World State Window, and the Player timeline. Each of these serves a different purpose with respect to conveying information about the Clip, and so handles the Markup hints differently. For example, while the Clip Editor renders all Markup hints, the Clip Browser only renders Markup for Clips that have been selected. The Player timeline ignores Markup, but shows a textual annotation for any Label that is attached to a Clip being displaye

3.6.6 Implementation Details

All of the CaPla tools have been implemented in Adobe Flex. Extensions are also implemented in Flex; we have implemented several to test the new mechanisms. Adding new Extensions is not burdensome. The Stops Extension, described earlier, requires only 123 lines of Action Script, and other similar extensions require a similar effort. More complex Extensions are possible as well. We implemented a Transportation Mode Extension that requires only 189 lines of code, however this does not include the applyTransform method, as this Extension only supports filtering and visualization. Note that the Extension code only represents the online portion of the Transportation Mode detection

algorithm. The model is built offline using CRF++,[16] implemented through the work (Y.-J. Chang & Newman, 2012).

While our study and examples up to this point have focused on location trace data, all of the features just described were intended and designed to generalize to other sensor data types. We have built renderers for generic one-dimensional time series data that we have used to experiment with traces containing accelerometer readings, and are in the process of developing Extensions for such data that address attributes such as mean amplitude, energy, periodicity, and inter-axis correlation (following (Bao & Intille, 2004a) and (Ravi, Dandekar, Mysore, & Littman, 2005a)). Many other sensor data types can be rendered as simple 1-D time series (thus graphed in two dimensions), such as RFID/Bluetooth tag sightings, audio, temperature, etc. We expect that most Extensions, however, will apply to only a subset of sensor data types, and our ongoing work includes the definition of a declarative syntax for matching Extensions to sensors.

While our study and examples up to this point have focused on GPS trace data, all of the features just described were designed to generalize to other sensor data types. We have built renderers for generic one-dimensional time series data that we have used to experiment with Clips containing accelerometer readings, and are in the process of developing Extensions for such data that address attributes such as mean amplitude, energy, periodicity, and inter-axis correlation (following (Bao & Intille, 2004b) and (Ravi, Dandekar, Mysore, & Littman, 2005b)). Many other sensor data types can be rendered as simple 1-D time series (thus graphed in two dimensions), such as RFID/Bluetooth tag sightings, audio, temperature, etc. We expect that most Extensions, however, will apply to only a subset of sensor data

[16] https://taku910.github.io/crfpp/

types, and our ongoing work includes the definition of a declarative syntax for matching Extensions to sensors.

3.6.7 The CaPla User Study

To see whether and to what extent the features of CaPla represent an improvement over RePlay and TraceViz with regards to supporting developers' needs when working with captured data, we conducted a user study. In order to facilitate comparison, we replicated the original RePlay study design as closely as possible. Specifically, we recruited Java developers, again used H&N as the sample application, began each session with a demonstration of H&N and RePlay, asked participants to improve the ETA and Arrival Detection algorithms, and debriefed participants on their experience at the end of the session. We recruited ten participants whose demographic characteristics and programming experience resembled those of the RePlay study participants (but not the same group of people). Data collection and analysis procedures were copied from the RePlay study as well.

The CaPla study setup differed from the RePlay in two regards. First, all sessions were conducted with single participant rather than mixing pairs and individual participant sessions. We did not feel that there was any benefit to having pairs in the RePlay study, so we opted to simplify this aspect. Second, we supplied participants with 200 location traces instead of 70, and the traces were not given descriptive names or organized into Episodes. We assumed that this change would make the tasks more challenging, and allow us to more fully test CaPla's new features, especially those related to finding relevant traces.

3.6.7.1 Study Results

Participants in the CaPla study experienced substantially greater success than did participants in the RePlay study. Applying the same criteria for success, i.e., the participant was able to modify the code for the given algorithm and articulate a coherent argument for why the changes represented an improvement, seven of the CaPla participants succeeded in both tasks, one more succeeded in one of the tasks (and did not attempt the other due to time), and two failed to complete either task satisfactorily. The two participants who failed both tasks did so primarily because they struggled to understand some aspect of the study setup. P3, for example, spent almost all of her time trying to understand how the windowed average ETA algorithm computed users' speed, whereas P10 never fully understood the relationship between CaPla and H&N, and tried to solve the ETA task by modifying her selected Clip rather than the H&N code.

Recall that only one of the RePlay participants was able to complete both tasks, and none of the other nine participants was able to complete either one. We attribute the greater success with CaPla principally to the reduced viscosity participants experienced when selecting traces and putting them to use. Based on think-aloud data, it was clear that the ability to view higher-level information about traces, such as speed and stops, contributed significantly to participants' ability to rapidly find useful data, as did the ability to focus on specific geographic locations through brushing. While we did not attempt to quantify the differences in time spent searching for data as compared to time spent improving and iteratively testing the code, it was clear that CaPla participants spent far less time searching for traces and far more time working with the code. Thus, we conclude that the facilities of displaying higher-level information about traces provided succeed in providing greater support for exploring and selecting examples.

CaPla's success in supporting other aspects of working with contextual data—modifying data and streamlining playback—was less clear based on this study. Only two participants attempted (successfully) to modify traces using the Clip Editor. One other participant grew frustrated that she could not find a good trace for testing her changes to the Arrival Detection algorithm. At the end of her session, it was suggested that she could have used the Clip Editor to create a Stop in the trace she had been working with, to which she replied that she wished she had remembered that it was possible to do that. Several other participants used the Clip Editor to view more detailed information about traces, but did not see a need to modify any data because they were able to find traces that they felt adequate for their testing needs. It is possible that the need to modify data did not appear as strongly in the CaPla study because of the greater number of traces (making it more likely that participants could find something suitable) and the relative ease of finding and trying different traces.

We did not find any evidence that CaPla's support for streamlined playback was beneficial to participants in the study, despite the fact that nearly all participants expressed impatience at having to wait for a trace to play through in order to find specific events such as a customer's arrival at a particular merchant's location. In several of these cases, automatically generated annotations would have provided valuable hints to speed up the process of finding the desired events, and more effective strategies could have been employed wherein the trace was rapidly explored and manually annotated, and then played more slowly in only the regions of interest. However, none of the participants discovered either of these strategies during their sessions, and it appeared that the Clip Player annotations were simply not noticed.

It is unclear whether the advanced capabilities of the Clip Editor and Clip Player were not noticed, not remembered, not needed, or not found useful in the form we

provided. However, several participants noted that they found CaPla to be quite complex, and admitted that they stuck to using a few features that they were able to understand and master in the short time available. Further study will be required to clarify whether and to what extent the added features, namely, trace editing and annotation-based playback, are useful, and also whether it is possible to decrease the complexity of interacting with CaPla while still retaining its benefits. During the debrief, P3 expressed the common sentiment that, while CaPla would have value for certain tasks within a development project, it would not be central enough to most projects to merit mastering the level of complexity it currently exhibits. He stated bluntly that *"I never want to see 200 Clips, I only want to see a few that are relevant to what I'm doing,*" suggesting that perhaps a capture-and-playback tool should take a more active role in ranking and recommending traces for particular needs. This remains a promising direction for future work.

3.6.7.2 Study Reflection

The results of the user study of CaPla validates the core functionality of the CaPla toolset and points out additional challenges that must be addressed to improve the integration of capture and playback into development. For example, the CaPla study shows that providing *semantic* controls with annotations for visualizing and selecting data has benefits *for improving the development process*. That is, not only do we replicate the TraceViz finding study finding that dynamic queries and location brushes are useful for helping developers find data, but that *helping developers find data leads to substantial improvements in their overall effectiveness*. This is shown by the observation that nearly all participants were able to complete tasks that they were unable to complete with RePlay, and that they were able to advance their understanding of both the data and the program code to a greater degree than they were without CaPla.

While CaPla was designed to also support the modification of captured data and streamlined playback for iterative testing, our study did not provide concrete evidence of the value of these features. It could be that the features are less valuable than suggested by the RePlay study or it could be that the toolset is too complex to master in a short study sessions. Perhaps the benefits are only realized over a longer time period, when developers become intolerant of the inefficiencies of playing long stretches of irrelevant data or when they wish to test specific failure modes that occur only rarely and are therefore difficult to capture in a natural setting.

We have used the CaPla tools internally for larger scale projects and plan to deploy them outside our group in the future in order to gain a better understanding of the long-term challenges and benefits of incorporating capture and playback across the development lifecycle. Taking a larger view of the capture and playback process will also allows us to address critical questions that we were not able to address through the controlled, limited studies such as the quantity and diversity of data required to address different types of concerns in context-aware system development; at what stages of development should data capture activities take place; and how can such data can be captured efficiently. In the previous sections, both TraceViz and CaPla were designed, developed, and evaluated in the situation in which data has already been captured. They are also designed particularly to support exploring, selecting, and playing back data, which are activities happening primarily in the *testing* process of context-aware systems.

3.7 General Discussion

3.7.1 The Benefits and Limitations of Capture-and-Playback

In this chapter, we have shown several examples of how C&P supports design and development of context-aware systems. In the two case studies, we have demonstrated the feasibility of playing back captured data for prototyping and evaluation. Specifically, in prototyping, C&P allowed us to examine different design alternatives, answer specific design questions, and identify design and usability issues with realistic data before evaluating it with real users. This helped reducing unnecessary distractions to participants and helped them focus on the user experience of the system. In evaluating the two projects, through playing captured data using the WOz approach, we were able to create realistic situations for participants to act out in study sessions. In the user study of RePlay, we also showed that developers were able to find suitable traces for testing the ETA and Arrival Detection algorithms. In addition, most of them could, at least, make sense and talk through the strength and weakness of the tested algorithms using the traces the chose.

On the other hand, we also learned that effective testing and evaluation greatly relied on the availability of behavioral data. For example, finding deficiency in the captured bus data made us compromise the design of the usability test tasks in the BusBuddy project. We had to design tasks for which our data could support. Unfortunately, we thought the data deficiency issue might be hard to avoid because in the early stage of the design it would not have been clear what kind of bus data we would need for prototyping and testing. As a result, we were not able to record those earlier. Although we had tried to collect those specific data we needed, we also realized that it was not always pragmatic to perform and collect those activities by ourselves, especially when those activities did not well fit the daily routine of the team members. Consequently, one important lesson we learned from the two case studies is that C&P may be more beneficial to design

and development of context-aware systems when the design team possesses abundant data or when it is pragmatic for the team to conduct targeted and specific data capture while the data are deficient. However, this might be an unrealistic aim for every design team to achieve. In our opinion, we think a more promising solution might be to build a platform on which the design team can access and request behavior data from a crowd of smartphone users, such as those existing crowdsensing campaigns (D'Hondt et al., 2014; Joki et al., 2007). We believe such a platform would enable design teams to more easily find data for different design and development activities, and thus help the teams move from "data constraining design" toward "data informing design." I will illustrate this proposal in more detail in Section 3.7.3.

3.7.2 Facilitating Identifying Good Examples of Data is Crucial

The most common yet challenging task we found in the case studies and the user studies was finding good examples of data for testing and evaluating the system. Examples sometimes were entire traces or regions in traces that showed a certain behavior (e.g. bus traveling along a certain street), displayed certain characteristics (e.g. speed, number of stops), or contained particular critical incidents (e.g. jumpy locations, signal lost, long dwelling events, entering the range of certain area). In other times, examples were scenarios involving multiple traces (e.g. multiple buses arriving around a transit center). Lacking examples of such, the design team would need to either capture data for those examples, creating one from existing data (e.g. modifying or transforming existing data, if applicable), or even modifying the original plan. However, even when these examples existed in the dataset, it remained a great challenge to locate and identify particular traces and regions among a large dataset. In our case studies, we attempted to accelerate searching examples by adding a descriptive name to each behavioral trace. Since people who collected data were not necessarily those used data later, adding these "annotations" helped the rest of the design team

members better understand and communicate the data. However, although adding descriptive file names post hoc, i.e. not during but instead after data collection, did help finding examples quicker, this method might be less efficient as the number of data traces grows. In addition, file naming is also not an effective way to reveal characteristics of traces such as speed of a location trace or of particular regions of a trace. Many participants in the user study had to play through an entire trace to find useful segments even after they read the description of the trace.

To address the challenge of finding examples, we built CaPla integrating TraceViz and RePlay to allow developers to see visualized characteristics of location traces and to filter location traces by directly brushing trajectories. The user study of CaPla then showed that both of these features substantially improvements developers' performance on finding examples for testing H&N. This conclusion was drawn from the observation that nearly all participants in the CaPla user study were able to complete tasks that participants were unable to complete in the first RePlay user study, and that they were able to advance their understanding of both the data and the program code to a greater degree than they were without CaPla. In addition, visualizing characteristics of traces (e.g. speed, location of stops) enabled participants to quickly locate and identify regions in traces useful for testing ETA and arrival detection, respectively.

However, we also think improvements can be made to help developers narrow down examples quicker, for example, highlighting or only presenting traces or regions relevant a specific testing or prototyping task. Continuing the proposal of building an online platform, we propose that the platform provides facilities of CaPla, and additionally provides recommendations of data traces based on the design activities, on the feature being tested, on the condition being examined, and on the review/ratings of data added by participants of the platform. We

perceive a great challenge of building such a behavioral trace data recommender system online, yet think it is a direction worthwhile to pursue in the long run because finding examples is a very common but essential activity during development.

3.7.3 Leveraging Mobile Crowdsourcing for Data Sharing and Requesting

As proposed, we argue that an online platform for people to contribute, share, access, and request behavioral data is a long term goal to pursue for facilitating the development of context-aware systems. This proposal is mainly derived for addressing the data deficiency challenge in using C&P for developing context-aware systems. The idea is that while the design or development team may not have enough people for collecting diverse as well as specific behavioral trace data, we believe a platform for requesting data collection from the mobile crowd may help resolve the issue. Specifically, in the early stage of the design project where opportunistic data collection is preferable because the features of the system may have not been determined, the mobile crowd can help collect more diverse behavioral trace data. While the design team has more specific need for data and meanwhile considers it not pragmatic to collect the data by the team members, the mobile crowd is also a valuable resource for requesting collecting specific data. For example, literature in mobile crowdsensing has shown the feasibility of requesting data collection from participants based on their mobility behavior and history (e.g. He, Cao, & Liu, (2015); Konomi & Sasao (2015); Sasank Reddy, Shilton, et al., (2009)). Using a similar approach, we propose the platform can help developers identify participants who might be suited for collecting certain behavioral data based on their behavioral history. Nevertheless, several open questions for such an approach remain: what constitutes a good instruction for this type of data request? What would be a good set of criteria for finding the suited participants? What would be a good approach for the mobile crowd to collect data? When would participants be receptive to a data collection

request and how do we find these moment(s)? How do we reduce participants' burden while requesting data from them? How do we know who contribute trustworthy and high -data? How do we assess the quality of participants' data? I will address some of these questions in this thesis in later Chapters, but seek to address the others in future research.

3.7.4 Limitations

The current chapter is subject to a number of limitations that need to be addressed in the future research of context-aware system development. First, the features of both RePlay and CaPla primarily deal with location and occupancy data type. As a result, the systems we developed and tested were all limited to responding to these two sensor types. Although we think that, for example, the three development activities—selecting examples, modifying data, and supporting playback we identified should be also essential activities in developing context-aware applications involving other types of contextual data, it is likely that different levels and kinds of challenges might emerge in these three activities when developing other types of context-aware applications. For example, while CaPla provides visualization and allows brushing trajectories for filtering location traces, it is reasonable to assume that different sensor types would require different presentation and visualization of sensor data and need different direct manipulation techniques for exploring, filtering, and selecting the examples. In addition, compared to sensor data such as accelerometer, proximity, rotation vector, light, and so on, location trace data is relatively more interpretable and is also more familiar to designers and developers because of its popularity. It is unclear, as a result, what different supports developers and designers would need for making sense of and for interpreting sensor data other than location. Furthermore, the context--aware applications we built are limited to receiving only one or two data types independently. It is likely that developers may want to develop context-aware applications responding to more data types or responding to an aggregation of multiple types of data (e.g. combining light sensor,

microphone, location, and accelerometer for distinguishing among places such as night club vs restaurant vs. home). Furthermore, CaPla are also primarily focused on applications responding to real-time contextual conditions and do not take previous contextual conditions into account. Thus, applications which consider previous user behaviors and conditions to determine how to respond to the current contextual condition may not be fully tested using the current features of CaPla. To support the development of such more complex context-aware applications, future work needs to answer: How do we present various types of sensor in the Clip Browser as well as in the playback tool to make sensor data more comprehensible and easier to explore? How do we support developers to modify different sensor data? What would be a good way to select examples of different sensor types? How would the data deficiency issue be different if we are to collect different types of data? Future research is needed to reexamine the results reported in this chapter in the context of developing more complex context-aware applications.

Finally, another important limitation of the two case studies is that we only executed one circuit of the design lifecycle for each project and that we drew lessons only from two design projects. Therefore, we were not able to know how the data capture and playback plan would change, what other challenges might occur, and whether designers and developers would need additional support in the second and further iterations, respectively. As developing context-aware applications become a more common practice among developers, it is worthwhile to reexamine to study results with more different types of applications with more iterations in future research.

3.8 CONCLUSIONS

Tools for capturing and for playing back sensor data are emerging in recent years. While making use of captured data for prototyping and testing context-aware

applications has been proposed in several previous design tools including RePlay, this chapter is the first to examine the process of working with captured data during development and to identify key challenges designers and developers face in using a C&P approach and a tool for developing context-aware applications. We present lessons learned from designing and developing two context-aware systems and provide key findings from a user study of RePlay. Specifically, we showed evidence that the C&P approach provides benefits to prototyping, testing, and evaluating location-aware systems. However, such a benefit is largely dependent on the availability of data. Because capture-and-playback is an iterative process and the need for and the usefulness of data may change based on the design activities, the design team is likely to encounter a challenge of having deficient data for playback, possibly leading to a situation of "data constraining design." To address this challenge, we propose leveraging mobile crowdsourcing to help capture both diverse and specific data, with the hope that this would help the design team have abundant data and move toward "data informing design." Nevertheless, when the design team has a large dataset for playback, one great challenge is finding good examples among the dataset for answering design questions, examining design alternatives, testing algorithms, and simulating realistic scenarios in user studies. It is especially difficult to identify specific useful regions within a trace. Annotations are particularly useful for addressing this challenge. While annotations help the design team gain a better understanding of available data, facilitate communication about data, and help searching examples quicker, system-generated annotations displaying characteristics of data traces are especially helpful for identifying useful segments within traces. However, despite annotations added during the review or by the systems have these benefits, annotations added during data collection in the field provide other valuable information, such as what behaviors the data traces represent, the context in which the data traces are collected, the and what critical incidents happened during data collection.

To support the design and development of context-aware applications in the long run, we argue for an online platform that equips features that CaPla provides and additionally allows designers and developers to access, share, contribute, and request data. While CaPla might be sufficient for internal use, building such an online platform can enable more designers, developers, and researchers interested in developing context-aware applications to have abundant data for prototyping, testing, and evaluation. As a result, such a platform also makes a larger scale study possible to investigate the development of context-aware applications with a larger variety of system and data types.

Chapter 4. Investigating Mobile Users' Ringer Mode Usage and Attentiveness and Responsiveness to Communication

4.1 Introduction

The growing adoption of mobile smartphones has dramatically changed the way we interact with computing technology and how we communicate with other people. The "always on, always connected" promise of mobile phones means that we can interact with information and with other people in an almost unlimited number of situations and contexts. This accessibility that the mobile smartphone enables also brings with it challenges in managing the potential for interruption and disruption. Just because we have a mobile device that is always on and always connected does not mean that we are always available for and aware of incoming communication. While mobile devices accompany their users most of the time, users only intermittently pay attention to their devices (Danninger, Kluge, & Stiefelhagen, 2006b), depending on where they are and what they are doing (Ferreira et al., 2014). Even when users are aware of or have attended to incoming communication, there are additional decisions of whether and when to respond (Avrahami et al., 2008).

Many computer-mediated communication (CMC) tools currently include availability signals (e.g., online, away, green, yellow, or red indicators) to help senders decide whether this is a good time to make contact. Research has shown that such a signal can help both senders and recipients coordinate communication (De Guzman et al., 2007; Schmidt et al., 2000). But these efforts are based on research largely focused on work-based office settings (e.g.(Danninger et al., 2006b; Fetter, Seifert, & Gross, 2011b; Fogarty et al., 2004)). Mobile devices

have brought availability and interruption issues into more diverse and unpredictable environments, making it harder to predict mobile users' availability for communication and harder to provide reliable and accurate signals of their availability.

To address this challenge, we seek to gain a better understanding of how mobile users manage interruption by and awareness of incoming communication in their daily lives, and how their attentiveness and responsiveness to incoming communication is influenced by such management practices. Specifically, we seek to understand mobile users' ringer mode usage, quantitatively and qualitatively, as it is the major function of mobile phones for managing the saliency of phone notifications. Then, we examine their attentiveness and responsiveness to incoming messages in different ringer modes and at different locales.

We conducted a two-week empirical study with 28 Android smartphone users. To collect their real usage of the phone for communication, ringer mode changes, and qualitative experiences, we employed a mixed methods approach including phone logging, diary study, interviews, and post-study survey. After reviewing related work, we describe the details of our study and design implications learned from it.

4.2 Related Work

While much of the prior availability research has focused on work office settings, some recent studies have focused on mobile platforms. Rosenthal et al. (Rosenthal et al., 2011) used an Experience Sampling Method (ESM) to acquire training data to develop a model for predicting phone interruptibility that automatically silences the phone when the user is uninterruptible. Pielot et al. (Pielot, de Oliveira, Kwak, & Oliver, 2014b) built a model to predict user's attentiveness to instant messages using features including user's interaction on the notification center, screen

activity, ringer mode, and sensors. Mihalic & Tscheligi (Mihalic & Tscheligi, 2007) explored how message type, mood, and communication channel and content affected how users would like to be notified of contact requests on their mobile phone. Fischer et al. (J. E. Fischer et al., 2010) used ESM to examine how content type and time of delivery affect receptivity to SMS interruptions, and concluded that the content of a message affects receptivity more than time of delivery. Poppinga et al. (Benjamin Poppinga, Heuten, & Boll, 2014) and Sarker et al. (Sarker et al., 2014) both used location, time, and sensor information to build a model for predicting opportune moments to deliver notifications/intervention tasks. Pejovic et al. (Pejovic & Musolesi, 2014c) used a similar approach with additional features, including emotions and engagement, to implement an intelligent prompting mechanism. While these previous studies suggested clues to predict when users would be interruptible, attentive, responsive, and receptive, respectively, our study builds on prior work to provide insights into mobile users' current practices of using ringer modes to manage interruption by and awareness of incoming messages and identify how ringer modes and locales affect their attentiveness and responsiveness.

Prior research has also explored sharing awareness and context on mobile devices. Ljungstrand (Ljungstrand, 2001) identified the need for sharing contextual awareness among mobile phone users to help judge their availability for receiving a call. Schmidt et al. (Schmidt et al., 2000) explored sharing context information to prevent inappropriate interruption, and De Guzman et al. (De Guzman et al., 2007) studied contextual information that helps a caller decide when to initiate a call. While these works explore sharing context among mobile users to coordinate communication, we focus on *understanding* mobile users' attentiveness and responsiveness to notifications.

Recent research has started investigating how mobile users attend to their phones, with a primary focus on notifications. Sahami et al. (Sahami Shirazi et al., 2014b) showed that mobile users generally attended to notifications within a minute, but important notifications such as of incoming messages were attended to more quickly. Pielot et al. (Pielot, Church, et al., 2014) obtained similar results, but further found that mobile users could attend to notifications within several minutes regardless of ringer mode. Ferreira et al. (Ferreira et al., 2014) identified users' micro usage (shorter than 15 seconds) on mobile phones and discovered that 60% of them were reportedly triggered by notifications. While their work documented why, how, and how fast mobile users attend to and deal with notifications, respectively, we explore how and why users are able to maintain this general attentiveness despite ringer mode, and further show how ringer mode and locale affect attentiveness and responsiveness more in depth. In addition, we focus on communication activities of any duration instead of only micro usages.

4.3 Research Methods

We used a mixed methods approach, including phone logging, user diaries, surveys, and interviews to understand mobile users' ringer mode usage and attentiveness and responsiveness to incoming communication, and to uncover reasons that prevent mobile users from reading notifications. We focused on applications for interactive communication, including phone; SMS texting; Mobile Messaging Apps (MMA), such as WhatsApp, Viber, Line, Facebook Messenger; Voice over IP (VOIP) calling; and video chat, such as Skype and Google+ Hangouts. Our data analysis on communication events narrowed in on SMS messages (details explained in the result section). The study was conducted from July through August 2013.

4.3.1 Study Procedure

We recruited Android users living in North America who had a full-time occupation. We posted recruiting messages on several online Android forums and Android user groups in social media. Participants were instructed to complete the entire study by running our Android Logger app on their phone over 14 days and were provided with a $75 gift card gratuity. We anonymized recorded contact names collected in the data by hashing the contact label and phone numbers.

However, since contact information was important for us to identify responding messages, after the 14-day collection period, we asked participants to provide user-defined labels (e.g., wife, friend, colleague) of their frequent contacts during the study period. After labeling their frequent contacts, participants were given links to visualizations that showed their daily phone use rhythm and frequent contacts with the communication media they used with each of those contacts. In addition, they were provided a heat-map that showed where communication activities were detected. On the map, they were instructed to add labels for "locales" of highly concentrated areas of activity. This allowed us to convert GPS coordinates and nearby areas into semantically meaningful locales for data analysis. A web-based, post-study survey collected their qualitative feedback and experience in managing ringer modes, communication activities, and phone notifications. Based on the data collected, we invited a subset of participants (14) for interviews, who received an additional $25 gift card.

4.3.2 The Android Logger App

We developed an Android logger app that: 1) monitored communication-related events on participants' phones, 2) captured a context snapshot when detecting a targeted event, and 3) delivered a daily diary for participants to provide more context about specific events. Logged events included sending and receiving outgoing SMS and MMA messages; and initiating, receiving, and ending phone,

VOIP, and video calls. To obtain the list of communication apps to monitor, we surveyed the top communication apps in the Android Market and asked participants to name all communication apps they used on their phone. In addition, we monitored ringer mode change events and actions demonstrating users paying attention to their phones, including waking up/unlocking the phone and acting on notifications, since mobile users can already preview the content of certain incoming messages through these two actions.

To detect events and actions on the phone, we used the Accessibility Service API in Android to monitor users' actions on their phones. The Accessibility Service broadcasts user events such as clicking, typing, swiping, notification viewing, and many others. This stream of data within the context of specific apps enabled us to detect exactly when participants received, attended to, and acted on notifications; composed and sent messages; and accepted and declined a VOIP call using a particular app.

When detecting an event of interest, the logger app recorded a context snapshot of the phone. The contextual information included location, activity recognition (provided by Google activity recognition service API ("Recognizing the User's Current Activity," n.d.)), sensors, network, calendar, phone status (ringer mode, screen on/off), and the currently running application. Activity recognition includes five states: still, tilting (significant change of angle relative to gravity), in a vehicle, biking, and on foot. One major challenge of logging context snapshots on mobile phones is balancing power consumption and recording accurate information (Lin, Kansal, Lymberopoulos, & Zhao, 2010). Because participants needed to run the logger app at all times for 14 days, we recorded contextual snapshots only when detecting a targeted event instead of continuous tracking.

Our diary aimed to obtain qualitative feedback around detected events, which included missed or declined phone calls, periods with unread notifications for

over an hour, and ringer mode changes. To obtain these inputs, we devised an event-based diary that included a list of questions based on the logged events in the past 24 hours for participants to respond at the end of each day. We did not deliver event-based diaries to participants at the moments when events were detected (known as an event-based ESM (Sunny Consolvo & Walker, 2003)) because our targeted events focused on missed notifications and communications. Although ESM studies are well known for capturing real-time and in-situ responses, when users are not available to respond to communication, they also cannot respond to an ESM questionnaire (Christensen, Barrett, Bliss-Moreau, Lebo, & Kaschub, 2003; Sunny Consolvo & Walker, 2003). In the event-based diary, we limited each question to no more than three randomly picked events logged within the past day to lower participants' burden. These events were listed in reverse chronological order (the most recent first) with a timestamp next to it, as shown in Figure 4.1. The events we asked included ringer mode changes, missed calls, and intervals where notifications were not read for more than an hour. Participants were asked to select the reason for the lapse in reading notifications and add more context. The diary also asked whether participants were interrupted by their phone and whether they missed a communication on their phone that day. By default participants received a diary notification at 9:30 PM. They could configure the delivery time to their preference, or directly open the diary whenever they wanted to submit it. Clicking on the notification brought them directly to an e-mail compose window to record and send in their responses.

4.3.3 Participants

We screened for participants that were over 18 years old, had a full-time occupation (including a couple of graduate students), used an Android smartphone for at least two months, did not have a substantial travel disturbance during the study period, and used the phone at least daily for texting and at least weekly for calling. We also attempted to balance the participants for gender and

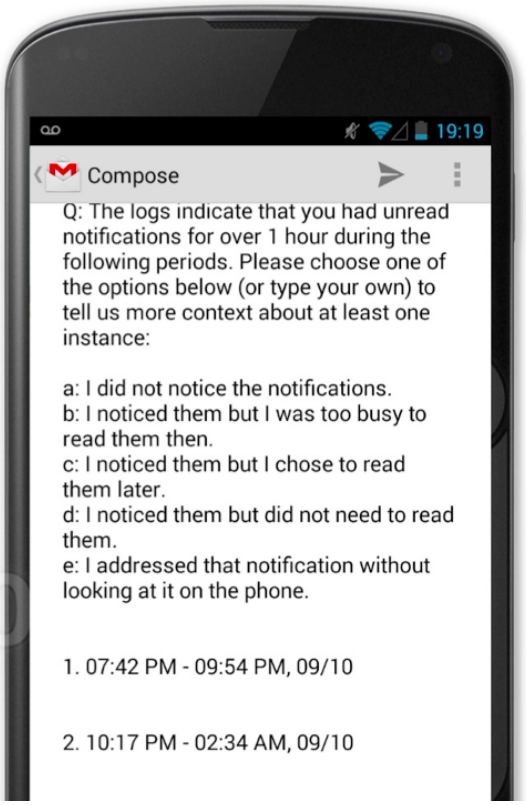


Figure 4.1 This daily diary question asks participants to provide more context about the periods when they did not read notifications for over an hour.

get a range of geographic areas and ages. All participants used an Android phone with version 4.0 or above (but below 5.0) during the study. While we started the study with 38 participants, 28 successfully completed the study (16 male, 12 female). There were several reasons why participants dropped out of the study: something happened to their phone, our logger system did not work accurately with their phone, or they did not comply with responding to the diary prompts. Most of the participants who successfully completed the study were in the 18-35 age range (25 out of 28). We refer to these participants as P1-P28 throughout this paper.

4.4 Data Analysis

Our analysis primarily focused on participants' availability and interruption management practices and their attentiveness and responsiveness under different ringer modes and locales. For the former, we mainly analyzed data from diary entries, survey responses, and interviews. We used descriptive stats on survey results; for qualitative feedback we reviewed diary entries and responses to open-ended survey questions and interviews, and open-coded them to identify recurring themes. In the survey and interviews, we learned about participants' overall strategies of and reasons for using each ringer mode. In the diary, we gathered 368 valid responses to ringer mode change events, where participants reported reasons why changes were made, and 832 valid responses to reasons for not reading notifications for more than an hour. We logged 1,107 ringer mode changes and analyzed them to look for patterns.

For attentiveness and responsiveness we only analyzed SMS messages because we found a disparity between incoming and outgoing MMA in our logs (shown later in the descriptive result). We expect that the main reason for this disparity is that our phone logger counted notifications of all incoming MMA messages, but participants might respond on another device (tablet, computer), which would not be counted in our log. Because this disparity would bias the result, we chose to only include SMS messages into the analysis of attentiveness and responsiveness.

To analyze attentiveness and responsiveness to incoming SMS, we grouped any SMS message between "the same contacts" within 6 minutes of each other together as part of a conversation. This allowed us to look at the message threads per contact, and know when a message is a response to the same person. The 6-minute threshold was chosen because prior work (Battestini, Setlur, & Sohn, 2010) found that the average time between text messages was 6 minutes. We also distinguished two scenarios: receiving *new messages* (i.e., no other message

within 6 minutes before the current message), and receiving *chat messages* (i.e., at least one message exchanged within 6 minutes before the current message). We separated them out because we assumed users' attentiveness and responsiveness to chat messages are higher than to new messages because they may expect to receive more incoming messages when they have been engaged in a chat.

As mentioned earlier, we logged *waking up/unlocking the phone*, *actions on notifications* (pulling down the notification bar and selecting notifications), and *composing outgoing messages in the same communication app that generated a notification*. These are three user-initiated actions that demonstrate paying attention to the phone in version 4 Android smartphones or above. We used intervals between these "attending actions" to the phone to measure *general attentiveness*, i.e., how often participants attended to the phone, and thus, how aware they were of the events on the phone. We computed intervals and compared among intervals using the 6-minute threshold, which were: <1 minute, 1-6 minutes, and > 6 minutes. We also measured specific attentiveness to notifications generated by incoming SMS to examine how promptly participants attended to the phone after receiving incoming SMS (referred to as *attentiveness to SMS*). We computed the intervals between receiving a notification of incoming SMS and initiating the first attending action after receiving that notification. For responsiveness, we coded whether an incoming SMS message was responded to with an SMS to the same contact.

For statistical analysis, we analyzed attentiveness as an ordinal dependent variable using mixed-effect ordinal logistic regression, with the categories: <1 minute (3), 1-6 minute (2), and > 6 minute (1). We analyzed responsiveness as a binary dependent variable using mixed-effect logistic regression. For both analyses we used *ringer mode* and *locale* as independent variables. We used mixed-effect regression because it allows us to add a random effect to separate out between-subject variance so that we could test the variables of interest. Similarly, we

analyzed ringer mode change using mixed-effect logistic regression, including *periods of day* as another independent variable.

In the sections below, we firstly present qualitative findings on participants' interruption and availability management practices, including self-reported ringer mode usage and strategies and reasons for not reading notifications. Then we present the quantitative results, mainly focusing on the effect of ringer mode and locale on attentiveness and responsiveness to incoming SMS messages, and on the effect of locale and time of day on ringer mode change.

4.5 Qualitative findings

4.5.1 Ringer Mode Usage

Ringer mode is a common feature of mobile phones for controlling signals of notifications the phone. In Android (before the latest Android 5.0), a phone both plays sounds and vibrates when the phone is in *Normal* mode. In *Vibrate* mode, sound is suppressed, but the phone still vibrates. In *Silent* mode there is no sound or vibration (but the screen or flashing light still activates). Because ringer modes directly affect how users notice notification signals, we sought to understand how participants used ringer modes to manage interruption by and awareness of incoming communication.

4.5.1.1 Self-Reported Ringer Mode Usage from Survey

Overall, our participants self-reported quite consistent strategies of using ringer modes for certain purposes. Most participants (23) reported in the survey that they put their phone in a quiet state (i.e. Silent or Vibrate) when they were sleeping, at work, or at occasions where they did not want the phone to interrupt them (e.g., spending time with family/friends, watching movie), and would return to the

mode where they could feel or hear notifications (Vibrate or Normal) afterwards to maintain awareness of notifications. *The other main usage of Silent mode was to prevent their phone from disrupting the environment*. 27 out of 28 participants reported that they had switched to Silent or Vibrate mode for this purpose. For example, P15 reported, *"When at work, my phone is in Silent mode so as not to disturb my coworkers."* P11 also explained, *"My ringer is usually on, unless I am receiving a lot of notifications, then I will switch to silent."*
The main reason for using Normal mode is to maintain awareness of notifications. S*ix* participants reported that they switched to Normal mode when they expected incoming communication, especially when they did not have their phone with them or were preoccupied with other things. As P5 stated, *"I would also turn my ringer on, so as to try to hear when someone was try to communicate with me. Since I don't always have my phone on my person at this location."*

Interestingly, we found quite diverse self-reported usage of ringer modes under these strategies. Figure 4.2 shows the amount of time users estimated that their phone was in each of the three ringer modes or turned off from the post-study survey. While some participants reported that they diligently switched between all ringer modes (e.g., P5, 13, 21), others switched mainly between two modes, or simply kept their phone mostly in one default mode. For example, P13 switched among all modes to manage awareness and to avoid phone interruption and disruption, whereas P17 reported that "99%" of the time he used Normal to keep high awareness of incoming communication. Overall, 16 out of 28 participants reported that they used one ringer mode more than 70% of the time. Only 3 participants had balanced usage of three ringer modes (all ringer mode usage > 25). The others mostly switched between any two of the modes.

Although this variation might have been because some participants more often encountered situations where they needed to silence their phones than the others,

participants also had different preferences and attitudes toward being aware of and interrupted by notifications. There were also different concerns about their phone disrupting the environment. For example, P22 reported that he almost always used Silent mode regardless of where he was: *"I feel I am in total control of when I want to see and handle notifications." He explained why he used Silent most of the time:"[I] don't like the sound of my phone going off. [J]ust a personal thing."* In contrast, P4 stated that he used Silent only 3% of the time: *"I really wouldn't use [S]ilent, … usually vibrate is my choice so at least I know something came that [I] can check later OR if [I] misplace it there will be some kind of sound from the phone."* P16 also claimed that she never put her phone in Silent mode because she did not need to*: "Any time I don't need it to be quiet."*
In summary, participants' self-reported strategies and purposes for using certain ringer mode seemed quite consistent. However, they chose different combinations of ringer modes for achieving their purposes due to their different preferences.

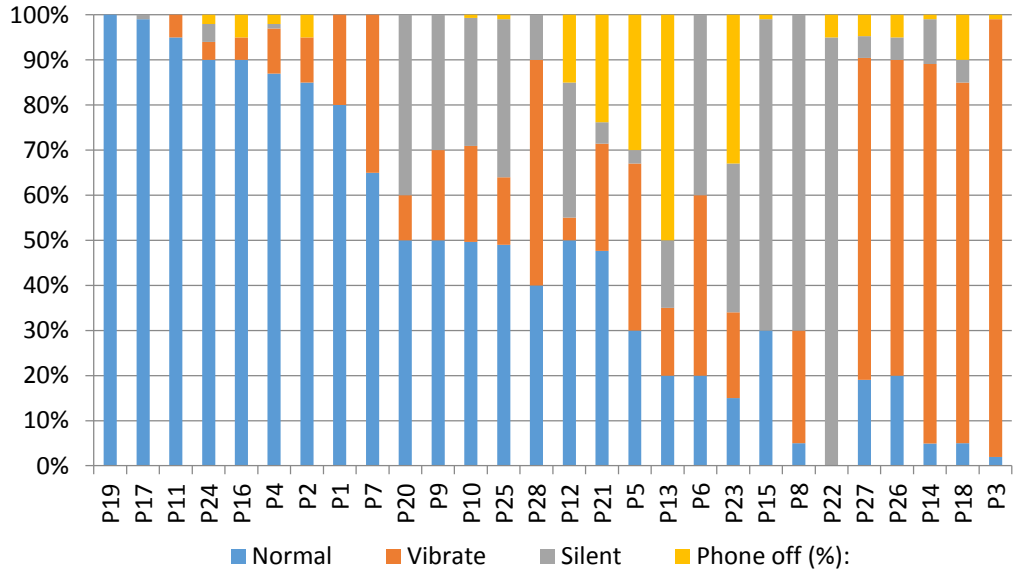


Figure 4.2 Comparing among participants' self-reported time their phones being in Normal, Vibrate, Silent, or off.

4.5.1.2 Reasons for Changing Ringer Mode from Diary

While the surveys gathered responses regarding overall usage and strategies of using ringer modes, the diaries uncovered actual reasons why participants changed to a certain ringer mode in their daily lives. We coded reasons from 368 responded ringer mode change events (Normal: 138, Vibrate: 133, Silent: 97) in the daily diary and reported on reasons frequently cited by participants.

Overall, the reasons cited in the diaries for using certain ringer modes were consistent with participants' overall impression in the survey results. The most frequently cited reason for changing to Silent mode were that they were going to bed (41 out of 97). Other frequent reasons included going to a meeting, being at work, and being in situations where the phone sound was interrupting and disrupting, such as watching a movie, being in a library or interview, or engaged in a chat. One typical response to these events is: "I was in a meeting and didn't want my phone to ring."(P13)

Reasons for using Vibrate mode were similar to using Silent mode. However, while many participants thought that Vibrate mode was sufficiently quiet, it also allowed them to notice notifications. As P8 reported in his diary: "I was going in to give blood and did not want to disturb anyone there but still wanted to be able to catch a call, text, or notification." P14 also gave a similar comment: "I was getting ready for work and wanted to be able to hear my phone go off then turned on vibrate because of work." This is perhaps why compared to Silent mode, Vibrate mode was a more popular option among our participants in their daily lives. In addition, very few participants changed to Vibrate mode during sleeping. This is perhaps when sleeping they cared less about incoming notifications.

The major reasons for changing to Normal mode were: enhancing awareness of notifications after leaving work, getting up from bed, and leaving environments

where phone sounds were considered disruptive. Other reasons included expecting incoming communication (mostly calls) and using mobile apps that require sound (videos, games, and navigation).

4.5.2 Reasons of Not Reading Notifications

We present reasons cited in the diary why participants did not attend to notifications for more than an hour. Figure 4.3 shows the breakdown of 832 valid coded responses. Over half of the notifications were not attended to because participants did not notice them (51%). Common explanations included that they were asleep (even though we tried to avoid asking them about intervals that occurred overnight), that the phone was inaccessible to them (in another room, charging), or that they were busy doing something else. For example, *"I will usually plug my phone in my room and leave it in there so will miss notifications for a while." (P9)*; *"was outside for swimming and didn't check my phone until afterward." (P23)*.

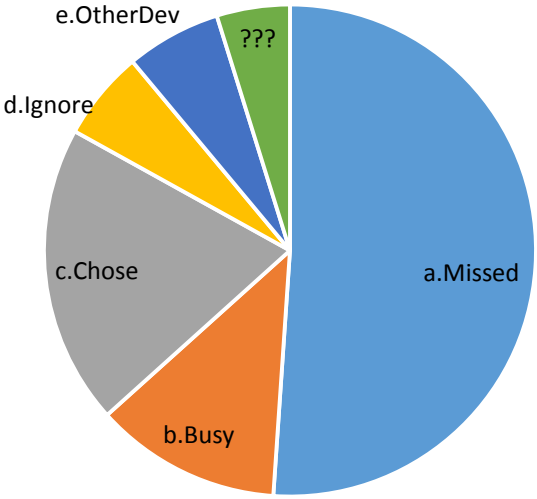


Figure 4.3 Figure 4.3. Responses for why users did not read notifications for over an hour: a) missed it, b) were too busy at the time, c) choose to read it later, d) ignored it, or other.

Other frequently cited reasons included noticing but choosing to address notifications later (19.7%), or being too busy to address them (12.3%) as illustrated by P12, *"Today was an extremely busy day with work; I didn't have time or energy to read the notifications when they first came." P8 also explained, "I did not want to handle them at the time but did not want to forget about them. [S]o I did not address them."* Sometimes, participants had addressed the notifications or will address them on another device (6.3%). For example, P14 said, *"I was also utilizing my tablet today and for the most part would get the notification on there and ignored them on my phone until later."* However, there were times where participants ignored notifications because they thought the notifications were unimportant (5.9%). *Sometimes they inferred this without actually checking their phone. P19 reported, "I was not really looking at my phone this evening, but most of these notifications were either unimportant [or] I addressed them on my computer."*

4.6 Quantitative Results & findings

Over the course of the study, we collected 11,986 incoming MMA (37.6%); 5,599 outgoing MMA (17.6%); 5,325 incoming SMS (16.7%); and 5,786 outgoing SMS (18.2%); Note that SMS and phone had an equivalent proportion of incoming and outgoing events, but incoming MMA was over 2 times the outgoing MMA. As mentioned earlier, we think this is because participants might respond on another device (tablet, computer), which would not be counted in our log.

4.6.1 Attentiveness to Incoming SMS

Figure 4.4a shows that participants' general attentiveness to the phone across ringer modes is quite similar. This result seems to agree with results recently reported by Pielot et al. (Pielot, Church, et al., 2014), which indicated that people

typically read notifications within several minutes regardless of ringer modes. We then focused on participants' attentiveness to incoming SMS. Regression results showed a significant effect of ringer mode on attentiveness for both SMS new and chat messages. For SMS *new* messages, both Normal ($p<.001$) and Vibrate ($p=.001$) are associated with higher attentiveness compared with Silent mode. Specifically, Figure 4.4b shows that the percentage of the attending actions within one minute in Silent (31.4%) was noticeably lower than in Vibrate (47.5%) and in Normal (44.8%), but the percentage for 1-6 minutes was not (Silent: 25.6%, Vibrate: 21.2%, Normal: 25%). This result suggests that without a notification signal (i.e., in Silent mode), participants were less likely to attend to their phone *immediately* after receiving incoming new SMS.

As to SMS chat messages, as expected, the attentiveness was much higher than the attentiveness to SMS new messages, as shown in Figure 3.4c **(< 1 minute**: Silent: 57.3%, Vibrate: 71.5%, Normal: 64%; **1-6 minutes**: Silent: 31.8%, Vibrate: 18.2%, Normal: 24.2%). We think this is because participants expected to receive more incoming SMS when they had been in a conversation. However, even with such an expectation, regression results showed that participants were still statistically significantly less attentive to SMS chat messages in Silent than in Normal ($p=.001$) and Vibrate ($p< .001$). According to Figure 4.4c, we believe this significant difference was mainly because participants were not able to attend to the phone immediately in Silent mode. After all, they were nearly equally attentive to incoming SMS chat within 6 minutes across all ringer modes (Silent: 89.2%, Vibrate: 89.7%, Normal: 88.1%).

These results together imply two things. First, the fact that participants have similar *general attentiveness* but achieve lower *attentiveness to incoming SMS* in Silent mode implies that in Silent mode participants' attending actions are not in reaction to incoming SMS (since they would not have any notification of

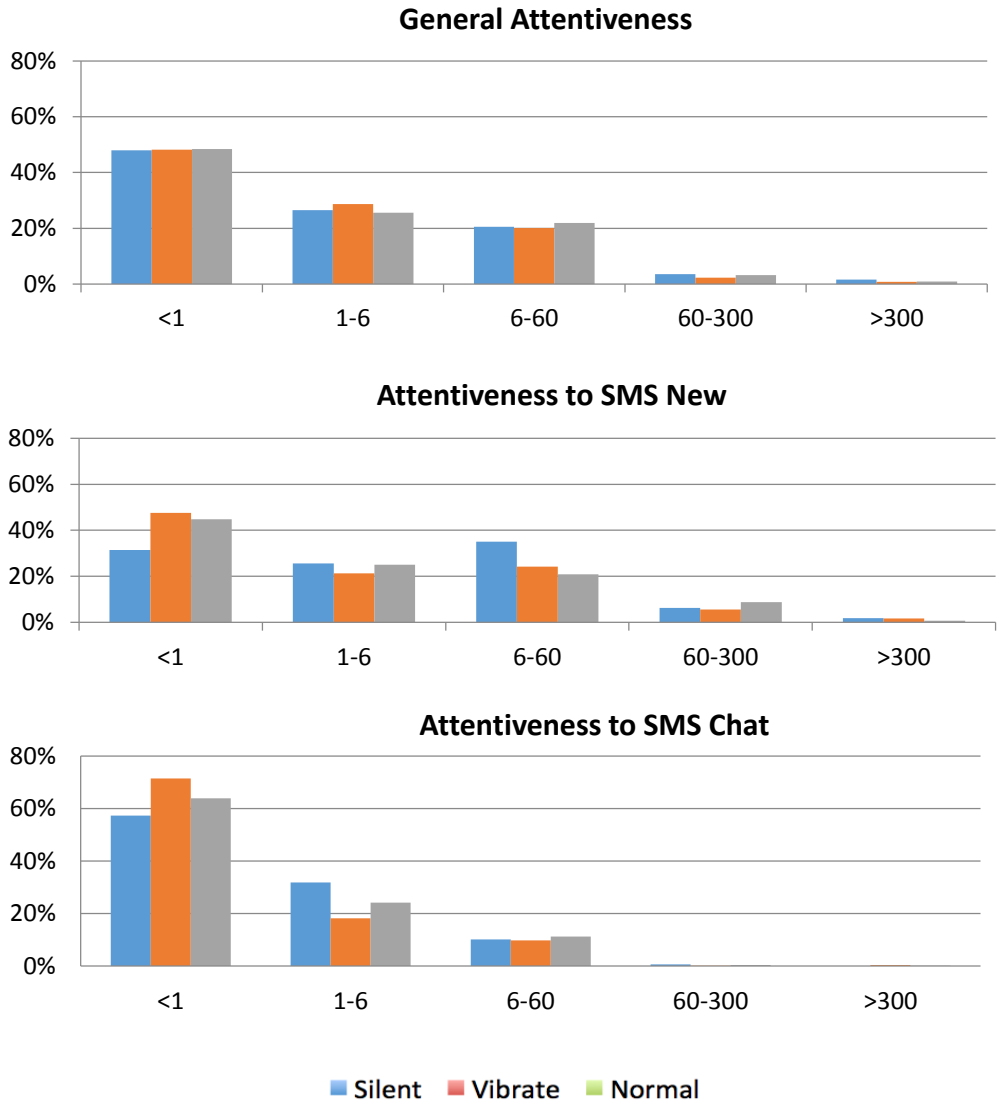


Figure 4.4 From left to right are: (a) intervals between attending actions, (b) intervals between receiving SMS new messages and the first attending action after it, and (c) intervals between receiving SMS chat messages and the first attending action after it.

incoming communication). Rather, the actions were distributed over time according to the participants' *spontaneous and proactive monitoring mechanism*. In contrast, in Normal and Vibrate modes, a notification signal (sound or vibration) evoked attending actions, suggesting a *reactive and notification-triggered monitoring mechanism*. Second, in terms of being able to immediately

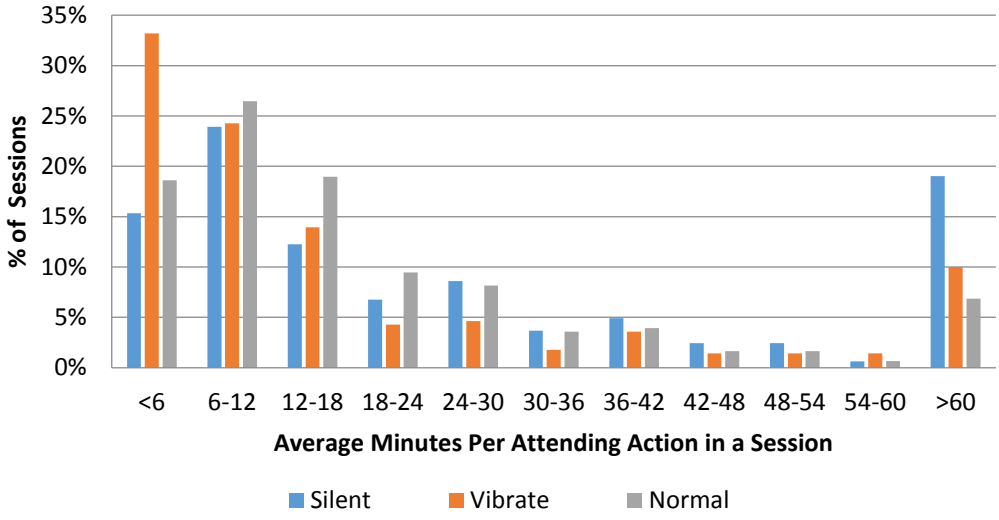


Figure 4.5 Average interval between attending actions in each ringer mode session.

attend to messages, the reactive monitoring mechanism seemed to be more effective than the proactive monitoring mechanism without a notification signal, even when participants might have developed expectation of receiving more messages.

Furthermore, we also examined attentiveness during *sessions* of ringer mode use. We define a *session* as *a span of time using a ringer mode until switching to a different mode*. For each ringer mode session, we calculated the average interval between attending actions within that session. Thus, a small value indicates that within that session a user frequently attends to the phone, and a large value indicates otherwise. Calculating the average attentiveness within a session allows us to see if participants uniformly checked their phones frequently (low average) within ringer mode sessions or not.

Figure 4.5 shows the distribution of average attentiveness in 749 sessions (Silent 163, Normal: 306, Vibrate: 280). It is interesting that there is a different pattern in

the distribution for Silent sessions compared to Normal and Vibrate. Silent sessions show a less steep decline as average attention interval increases, and actually shows a larger relative increase in sessions with an average longer than 60 minutes. There appears to be more variation in attention intervals for Silent sessions compared to Normal and Vibrate. This difference in pattern could be explained by two different reasons for silencing their phones. As demonstrated earlier, participants continue to proactively monitor their phone in certain situations when in Silent mode. Figure 5 shows a substantial number of Silent sessions with a short average attention interval. This likely represents times when the user is silencing their phone to avoid disrupting the environment, but still wants to maintain awareness of incoming communication. However, compared to Normal and Vibrate, there is a relatively higher proportion of Silent sessions with longer average attending intervals. This likely represents sessions where the users silence their phone to suppress notifications and avoid being interrupted. Thus, our data suggest the possibility of distinguishing when users silence their phones to avoid disrupting their environment versus interrupting themselves by tracking how often they continue to attend to their phone when in a Silent session.

In terms of the effect of locale (see Table 4.1), we used the Home locale as the reference group in a regression analysis. The results show that the Catch-up locale is statistically significantly associated with lower attentiveness both to SMS new ($p=.04$) and SMS chat messages ($p<.001$). Catch-up locale referred to places where participants did not spend large amounts of time, but frequently used their phones to catch-up on past incoming communications, such as train stations, parking garages, and regular lunch walks. Finding low attentiveness at Catch-up locale, especially within one minute, is unexpected because, according to our data, 74.6 % of the intervals between attending actions at catch-ups were within 6 minutes (< 1 minute: 59%, 1-6 minute: 15.6%). The high general attentiveness at Catch-up locale may be a side effect of being generally active on the phone.

	Home	Work	Catch-up	Social	Other
Attentiveness					
New SMS					
< 1 minute	45.8%	45.0%	25.9%	41.7%	52.6%
1-6 minutes	25.4%	19.4%	18.5%	22.3%	15.8%
> 6 minutes	10.1%	3.6%	25.9%	8.7%	7.9%
Chat SMS					
< 1 minute	69.5%	66.0%	33.3%	66.5%	62.8%
1-6 minutes	21.5%	22.4%	25.9%	21.1%	20.9%
> 6 minutes	9.0%	11.6%	40.8%	12.4%	16.3%
Responsiveness					
Attended New SMS	56.7%	65.4%	75%	65.2%	50%
Attended Chat SMS	80.7%	73.9%	68.8%	84.7%	63.9%

Table 4.1 Attentivenss to SMS new and chat messages and responsiveness to already attended messages by locales

Perhaps the highly mobile, transitory nature of Catch-up locale means that users are not able to address notifications immediately after receiving them.

The regression results also showed that attentiveness to SMS chat messages at "Other" locale is *statistically* significantly lower (p=.01). The Other locale referred to places where participants visited frequently during the study period, such as gyms, grocery stores, bookstore, etc. It seems that at these places participants were less able to chat. This might be because at these places they usually had a goal to accomplish and were engaged in certain activities.

4.7 Responsiveness to Incoming SMS Messages

Participants' overall response rate to SMS new messages was 39.1%, and to SMS chat messages was 70.3%. Since participants might be unresponsive because of they were inattentive to incoming SMS, we are primarily interested in their responsiveness to messages to which they had attended. The results showed that

	Silent	Normal	Vibrate	Overall
Attended New SMS	56.7%	53.6%	69.8%	57.5%
Attended Chat SMS	75.0%	80.6%	75.0%	78.2%

Table 4.2 Responsiveness to already attended SMS new and chat messages by ringer mode

participants' response rate to already attended SMS new messages was 57.5%, and to already-attended SMS chat messages was 78.2%, which, expectedly, were both higher than the overall response rate. However, this shows that still about 40% of messages to which they attended but not responded within 6 minutes.

Interestingly, while Table 4.2 shows that participants seemed to be most responsive to attended SMS new messages in Vibrate mode and most responsive to attended SMS chat messages in Normal mode, regression results did not show any statistically significant difference among ringer modes in responsiveness to either type of SMS messages at the 0.05 level of significance. This suggests that participants' responsiveness to incoming SMS differed likely because they were differently attentive to incoming SMS: once they were able to attend to a message, they did not significantly differ in their responsiveness in different ringer modes.

When investigating the effect of locales on responsiveness using the Home locale as a reference group, regression results showed that participants were statistically significantly less responsive to attended SMS chat messages when they were at the Other locale ($p=.003$) and at the Work locale ($p=.02$). However, the results did not show any statistically significant difference among locales for attended SMS new messages. These together suggest that participants seemed equally likely to respond to an SMS new message once they had attended to it at different locales. However, they were less likely to get engaged in a continuous conversation when

they were at work or were at places where they were often preoccupied by other things.

4.8 Ringer Mode Switches by Locales and Time of Day

We also investigated whether participants' ringer mode changes were associated with any locale and time of day. We logged in total 1,107 ringer mode changes (avg: 39.5, med: 27; max: 159; min: 3; std: 38): 475 were to Normal mode, 379 were to Vibrate mode, and 253 were to Silent mode. In particular, the majority (53%) of switches were between Normal and Vibrate modes (Vibrate to Normal: 290; Normal to Vibrate: 283); 29% were between Normal and Silent modes, and only 18% were between Vibrate and Silent modes. These results were consistent with participants' self-reports that overall participants more often used Vibrate as a *quiet mode* than Silent mode. Perhaps participants thought Vibrate mode was quiet enough and meanwhile allowed them to notice notifications. We plotted the distribution of ringer mode changes by locale and hour of day to look for distinct patterns of ringer mode switches. This allowed us to group hours into periods for a logistic regression analysis. We found several distinct patterns when ringer mode changes occurred. Regression results showed that changes to Silent mode were statistically significantly more associated with the Home locale from 9pm-2am ($p<.001$), which is likely linked to going to bed. Secondly, changes to Vibrate mode were statistically significantly more associated with the Catch-up locale ($p<.001$). Thirdly, changes to Normal mode were statistically significantly more associated with 4pm-6pm ($p=.008$), perhaps corresponding to getting off work or the commute from work. In addition, switches to Normal mode were also statistically significantly more associated with the Other locale ($p=.008$), perhaps at these places when participants were engaged in other activities, they wanted a more salient signal to notice notifications.

4.9 Discussion

4.9.1 Learning the Purposes behind Ringer Mode Uses

Our participants' self-reported quite diverse ringer mode usage. Based on their responses from the survey and the diary, the diversity was not merely because they were exposed to different contexts, but also they had different preferences of ringer modes and attitudes toward being aware of notifications and being disrupting the environment. Because of such diversity, in similar contexts participants used different ringer modes, and that some participants kept their phone in one ringer mode across different contexts, creating a challenge of inferring people's attentiveness/responsiveness primarily based on the ringer mode in use. Although we identified several patterns of ringer mode changes, those patterns only represented a small portion of a day. Furthermore, participants put their phone in the same ringer mode for different reasons, in which they might display different attentiveness and responsiveness. For example, while sometimes participants used Silent mode for avoiding interruption, at other times they wanted to prevent their phone from disrupting the environment. These together indicate that ringer modes themselves may not be a reliable signal of mobile users' attentiveness and responsiveness.

However, we found that a more reliable signal is the *purposes behind ringer mode uses*. Based on our findings, there are at least three purposes that can be distinguished: 1) for avoiding interruption, 2) for avoiding disrupting the environment, and 3) for noticing important notifications. For the first, users prefer not noticing a notification; they would set ringer mode in a way that the phone does not distract them. For the second, users mainly want to minimize the saliency of notifications *for the environment*—users themselves may still want to aware of the notification. For the third, users want to make notifications more noticeable for themselves, usually because they are expecting certain notifications. Users may use different combinations of ringer mode for achieving

these purposes, depending on their preferences (in our study, a popular combination was between Normal and Vibrate modes); however, we believe that these three purposes are useful signals of their current or upcoming attentiveness or responsiveness compared to ringer mode per se, especially when users just have switched a ringer mode. We also found a few patterns of ringer mode switches associated with certain locales and periods. We think it is worth associating the purposes behind ringer mode uses with locales and periods, perhaps creating *personas* representing common patterns of ringer mode use for designing future notification services.

One advantage of learning purposes and using them as an indicator is that they are presumably persistent and are independent from the features of mobile systems, whereas ringer modes vary on different systems and may evolve overtime. Moreover, once wearable devices become more pervasive and affordable, more mobile users are likely to attend to notifications across multiple devices. Focusing on why and when users want to avoid and to be aware of notification allows designing a notification service without being limited to any mobile system. We provided an example of computing the intervals between attending actions in Silent sessions to distinguish purposes of using Silent mode. Future research can devise more sophisticated heuristics to learn and distinguish the three purposes.

4.9.2 How are Ringer Modes and Locales Related to Mobile Users' Attentiveness and Responsiveness

We analyzed logs to investigate attentiveness and responsiveness in different ringer modes and locales. Our results provide a number of implications. First, while two recent studies showed that mobile users generally attend to notifications within several minutes, especially for those from communication apps (Pielot, Church, et al., 2014; Sahami Shirazi et al., 2014b), our results showed that, in terms of being able to *immediately* attend to a message,

participants were less attentive in Silent mode than in Normal and Vibrate modes. We believe this difference was not because they were less interested in notifications in Silent mode (given that they had a similar distribution of attending actions across all ringer modes, as shown in Figure 4a), but because without signals of notifications it was difficult for them to notice notifications immediately, even if they might have developed expectations of receiving more messages in a chat. This reason may explain why participants more often chose Vibrate mode as *the quiet mode*, as it allowed participants to more likely to notice notifications. One suggestion we have is providing additional undisruptive but noticeable signal for users (e.g., visual feedback). If a notification service can *infer* users' purposes for using ringer modes, providing multimodal signals of notifications whenever appropriate is generally useful (note that the top reason for not reading notifications is "not noticing them.") Secondly, our results did not show statistical significant difference in responsiveness to already attended incoming SMS across ringer modes. This implies that ringer modes mainly affect attentiveness but not responsiveness to already attended messages: once users are able to attend to a message, being responsive or not is less correlated with which ringer mode they use. We think this perhaps that diverse ringer mode usage weakens the relation between context and the use of a particular ringer mode.

Thirdly, we found an influence of locale on attentiveness and responsiveness. Participants were less attentive to incoming SMS at Catch-up locale, showing the highly mobile and transitory nature of these locations. They were also less responsive to incoming SMS chat messages when they were at the Work locale and at the Other locale. We believe at these places, because participants were often preoccupied, they were less available for continuously chatting. In addition, participants also self-reported their own individual behavioral pattern of ringer mode use at different locales (e.g. not reading notifications at church or at social

places). Taken these together, we believe that locale is useful information for inferring attentiveness and responsiveness.

Recently, Pielot et al. (Pielot, de Oliveira, et al., 2014b) showed that it is feasible to predict attentiveness using various features including ringer modes and screen activities. We think it is also worth exploring how the locale information, converted from GPS coordinates through users' input, improves the prediction. In addition, it may be also worth including "purposes behind ringer mode uses" as a feature for predicting attentiveness and responsiveness. One remaining challenge is predicting responsiveness as it involves an additional users' decision i.e. whether to respond. Although we identified the effect of locale, we believe another factor is *who* sends the message (Rosenthal et al., 2011), which, unfortunately, was not examined in this paper. Once estimated attentiveness and responsiveness become reliable and acceptably accurate, we propose providing this information for message senders, as it can signal senders when a good time would be to make contact. It would be interesting to explore this concept in a working prototype.

4.9.3 Implications for Requesting Data Collection from Smartphone Users

Despite the fact that the current study was not specifically aimed to address research questions in data capture and playback, we think several findings in this study provide implications for data capture—specifically, finding opportune moments for sending data collection requests to the crowd of smartphone users. First of all, the study results suggest that it is important to take ringer mode into account when sending a data collection request to smartphone users, because ringer mode would affect how quickly people can attend to communication requests. For example, while users' phone is put in the Silent mode, researchers may expect that users are less likely to immediately notice the notification. Therefore, if the researchers are sending urgent requests and expect users to

quickly respond to the notification, they may want to send the request to people whose phone is not put in the Silent mode. Furthermore, it would be worth studying users' general attentiveness pattern in the Silent mode to understand whether the users' current use of the Silent mode is for avoiding disrupting the environment or for avoiding being interrupted by the phone. If the users are found in the latter situation, researchers should avoid sending the request to the users. Furthermore, the results also indicate that it would be important to measure attentiveness and responsiveness respectively when investigating opportune moments for sending data collection requests. For example, the study results suggest that ringer mode and locale have different impacts on attentiveness and responsiveness. It is likely that the impacts of other contextual factors on attentiveness and responsiveness may also differ. Being able to complete a data collection request entails both attending to and responding to the request; failing to reach each step leads to the task request unfinished. Therefore, to better understand the actual reason for which users do not complete the task, it would be important to measure attentiveness and responsiveness respectively for each unfinished request and investigate what contextual factors contribute to inattentiveness or unresponsiveness to the request.

4.9.4 Limitations

The study presented in this chapter is subject to several limitations. The first limitation is regarding the applicability of the study findings. Our study focused on attentiveness and responsiveness to incoming communication on mobile phones. Thus our findings may not apply to other devices such as computers and tablets. In addition, we conducted the study on Android phones, and it is likely that iPhone users may display different attending and responding behavioral patterns from Android users. Furthermore, although we argue that the study results provide some implications for finding opportune moments for sending data collection requests, it is however unclear whether mobile users would display

similar attending behavioral patterns for notifications sent by the researchers, and what contextual factor would affect users' decision regarding whether to respond to the received data collection task. Second, we were not able to analyze MMA messages because several of our participants have multiple devices for MMA communication. Thus our analysis of attentiveness and responsiveness to SMS may not apply to MMA. Second, to reduce participants' burden, we only asked participants to provide labels of frequent contacts, and did not ask them to name the closeness with each of them. Although we could have inferred it through contact labels (e.g., spouse, best friend), that inferred information might not be reliable. As a result, we were not able to examine the impact of contacts on attentiveness and responsiveness. Third, we could only measure user attentiveness through users' related actions on the phone, as other previous work has done (e.g. (Pielot, Church, et al., 2014; Pielot, de Oliveira, et al., 2014b)), but we did not have the ground truth of whether they *actually* read each of the messages. Fourth, we could not reliably estimate the duration of using each ringer mode because we did not capture the information of when the phone was off. In future work, it is worth capturing this information to estimate the duration of each ringer mode to get a complete picture of mobile users' ringer mode usage in their daily lives.

4.10 Conclusions

In this paper, we investigated how mobile users use ringer modes to manage interruption by and awareness of incoming communication, and how that practice and locale affect their attentiveness and responsiveness. We highlight that mobile users have diverse ringer mode usage, but they switch ringer mode for three main purposes: 1) avoiding interruption, 2) preventing their phone from disrupting the environment, and 3) noticing important notifications. We suggest future notification services be designed for these three purposes. We also highlight that ringer mode mainly influences attentiveness but not responsiveness to attended messages. Without signals of notifications users are less likely to immediately

attend to SMS messages than with signals. In addition, mobile users are less attentive and responsive to SMS at certain locales. We suggest CMC tools learn to infer the purposes for using ringer modes associated with locales, and use them as features for building predictive models for attentiveness and responsiveness. This benefits not only CMC tools, but also researchers attempting to identify opportune moments for sending notifications and tasks to mobile users, such as sending data collection tasks. However, to explore opportune moments for requesting collecting behavioral data from the mobile crowd, future work must investigate mobile users' attentiveness and responsiveness in the context of sending data collection requests to the mobile crowd.

Chapter 5 An Investigation of Using Mobile and Situated Crowdsourcing to Collect Annotated Travel Activity Data in Real-World Setting

5.1 Introduction

The design of context-aware systems has been a topic of long-standing concern in the HCI and Ubiquitous Computing communities (Abowd et al., 1999; Chen, Kotz, & others, 2000).. Researchers and practitioners in these fields are seeking to develop systems aware of users' context and activity, thereby providing relevant information and/or services to the users. A common practice in context-aware system development is collecting contextual data representing user activities and contextual conditions that the system is expected to encounter when they are deployed in the field (Newman et al., 2010). Such data are needed for training and evaluating recognizers that detect important contextual states and trigger system responses (Dey et al., 2001), and are also important for use in the prototyping and evaluation stages of system development (MacIntyre et al., 2004; Newman et al., 2010). An essential step in collecting these activity data is to collect labels and annotations describing the data. These metadata not only allow developers to train and test their recognizers but also enables them to more easily filter and select suitable sets of data for testing the functionality of the system.

Researchers have used different ways to collect annotated contextual and activity data, including recording and annotating data on their own (DeVaul & Dunn, 2001), and using a structured participant-based approach, i.e. recording and annotating data with a small sample of people performing predefined activities in a controlled environment under the researchers' guidance (Bao & Intille, 2004b; Kwapisz, Weiss, & Moore, 2011). As sensor-laden smartphones have become pervasive, researchers have started exploring ways to leverage larger numbers of

mobile smartphone users—sometimes referred to as the mobile crowd—to record and annotate targeted activities using their smartphones in real-word settings (Abowd et al., 1999; Y.-J. Chang, Hung, & Newman, 2012; Newman et al., 2010). Two broad approaches are commonly used for crowd-based data collection and annotation: P*articipatory* data collection (Ganti et al., 2011; Kanhere, 2011) refers to the process in which mobile users actively participate in collecting data; they manually control an instrument to collect data based on their interpretation of researchers' needs and instructions (Paxton & Benford, 2009). *Opportunistic* data collection (Ganti et al., 2011; Lane et al., 2010) refers to the process in which the instrument automates the data recording: the mobile users carry an instrument that records data itself based on a certain sampling heuristic, where the sampling can be continuous, randomized, schedule-based, or context-triggered (Froehlich, Chen, Consolvo, Harrison, & Landay, 2007c; Meschtscherjakov, Reitberger, & Tscheligi, 2010). To obtain users' annotations, instruments can be programmed to prompt users to annotate during the activity being recorded to obtain *in situ* annotations, or to prompt users afterward to obtain *post hoc* annotations. Using these methods to collect contextual and activity data via the mobile crowd in real-world settings has a considerable advantage compared to the controlled data collection method: the collected data are more diverse, naturalistic, and representative to users' real life behaviors. However, because the data collection is not under researchers' supervision, it is also difficult to assure the quantity and quality of the data. At present, we have limited understanding of which approaches can reliably and effectively produce high quantity and quality of data, and this fact in turns limits the usefulness of mobile crowdsourcing for collecting activity data.

Due to this limitation, in recent years, research has started assessing the quality of labeled activity data collected in the field. For example, Cleland et al. (Cleland et al., 2014) showed that collecting labeled physical activity data in the field using a

Context-Triggered Experience Sampling Method (ESM) approach obtained equally accurate labels compared to those obtained in a controlled lab study. However, in this study, the authors neither analyzed the quantity and quality of activity recordings nor analyzed users' experience and behavior in using the approach. In addition, the controlled lab studies they compared were not performed in real word settings, meaning that Context-Triggered ESM was the sole approach being performed in the field. Thus, it remains unclear whether or not the Context-Triggered ESM approach is a more reliable and effective approach for collecting activity data compared to other approaches such as the Participatory approach. The purpose of this paper is to compare the effectiveness of different approaches in mobile crowdsourcing to collecting annotated activity data, as well as to understand participants' behavior in using these approaches so that we will gain a better understanding of how to design better tools and approaches for supporting mobile crowdsourcing to collect annotated activity data.

In this paper, we report findings from a two-week field study involving the mobile crowd comparing three approaches to collecting annotated travel activity data with mobile users in real-world settings, namely, Participatory, Context-Triggered In Situ, and Context-Triggered Post Hoc. These approaches were performed by 37 smartphone users to collect their individual travel activity data when they were traveling outdoors using our instrument. To obtain the ground truth of their travel activity during the study, we asked the participants to wear a wearable camera all day during the study and collected their location and activity traces. We also asked them to make daily entries into a diary to capture their challenges and errors made using the approaches, and conducted post-hoc individual interviews to understand their overall experiences, strategies, and preferences with respect to each approach. Moreover, we collected logs of the participants using the instrument to collect activity data. This allowed us to capture their actual behavior

while performing each approach beyond the recalled behaviors or subjective impressions of interaction obtained from the interviews.

We conducted two phases of analysis on the dataset. In the first phase (Phase One), we compared the quantity and quality of collected data among the three approaches as well as participants' subjective experience in using each approach. This allowed us to understand the pros and cons of each approach for collecting activity data and the aspects users value in activity data collection. Our results provide two highlights: first, the data collected using the Participatory approach were more complete, contained less noise, and led to greater data coverage than those collected using the Context-Triggered approaches. Second, while participants appreciated automated recording and reminders for convenience, they highly valued having control over what and when to record and annotate. As a result, we conclude that user burden and user control are two important aspects a future tool in mobile crowdsourcing should take into consideration. In the second phase (Phase Two), we extended our work by adding an analysis of participant behavior using the participants' behavioral logs. That is, we investigated how participants used our instrument to perform the approaches in the field to collect activity data and examined how the specific nature of the activities being captured affected their behaviors in collecting the data for those activities. Analyzing participants' behaviors, as (Dumais, Jeffries, Russell, Tang, & Teevan, 2014) has suggested, enabled us to obtain a more complete and accurate picture of participants' behaviors and patterns that they would have not been able to remember and articulate accurately. It also helped us understand any systematic biases in the data and to suggest ways to address them through the design of tools or methods. In addition, we also analyzed the characteristics of participants' annotations to understand whether annotations would differ according to the type of activity being collected, and analyzed the diary entries to understand the reasons for unlabeled, mislabeled, and erroneous data. In this analysis, we found

that the type of activity being captured influenced the timing of recordings and annotations, participants' receptivity, and characteristics of annotations. Moreover, these factors were impacted by the nature of transitions between activities, the attentional requirements of each activity, and the context of the activity. Based on the findings, we provide a set of design and methodological recommendations regarding the approach, tools, and instructions for using mobile crowdsourcing to collect activity data.

The remainder of the paper is organized as follows: We discuss related work in Section 2. We present the field study and explain our research methods in Section 3, then describe our general data processing and coding process in Section 4. We describe the analysis and present and discuss the findings of Phase One and Phase Two in Section 5 and 6, respectively. Then in Section 7 we provide a general discussion, including the design and methodological implications and the study limitations. Finally, we conclude in Section 9.

5.2 Related Work

5.2.1 Leveraging the Mobile Crowd to Collect Data

Leveraging a crowd of workers to perform tasks in the mobile environment has been gaining attention in recent years because of the wider availability of smartphones and mobile Internet. Since most modern smartphones are equipped with various sensors, many researchers have attempted to develop applications and platforms to collect sensor data from smartphone users, a method known as mobile crowdsensing (Ganti et al., 2011; Khan et al., 2013; Lane et al., 2010), and citizen science (Silvertown, 2009). Participatory Sensing (Kanhere, 2011), in particular, is a well known and widely used approach to collecting sensor data in the wild in mobile crowdsensing (Ganti et al., 2011; Khan et al., 2013; Lane et al., 2010). The idea of Participatory Sensing is that participants initiate data collection with guidelines provided by task requesters (usually researchers) and use an

instrument to capture data of interest for data requesters. Because researchers need to rely on participants to cooperate and to provide good quality data, much of prior research in Participatory Sensing focused on supporting participants, including protecting participants' privacy (De Cristofaro & Soriente, 2011; Ganti, Pham, Tsai, & Abdelzaher, 2008; Sakamura et al., 2014), reducing participants' effort by requesting data only from those who are in relevant locations (Linnap and Rice, 2014) or are moving to the target area (Konomi & Sasao, 2015), and improving the data quality (K. L. Huang, Kanhere, & Hu, 2010b; Sasank Reddy, Burke, Estrin, Hansen, & Srivastava, 2007; Sheppard, Wiggins, & Terveen, 2014).

Mobile and situated crowdsourcing, an emerging area that aims to overcome the limitation of online crowdsourcing on performing tasks beyond the desktop, is not limited to collecting sensor data. For example, (Goncalves et al., 2014) used public displays as a crowdsourcing platform to gather keywords to describe locations; (Heimerl et al., 2012) used a vending machine for performing locally relevant tasks; (Agapie et al., 2015) involved local workers to report local events. (Hosio et al., 2014), on the other hand, used a kiosk to offer a variety of crowd tasks, including typical crowdsourcing tasks such as identifying and annotating objects in images (Nowak & Rüger, 2010) and videos (Vondrick, Patterson, & Ramanan, 2012). However, the fact that the kiosk is deployed in the field allowed the workers to perform field tasks such as describeing the environment.

However, most, if not all, of these mobile crowdsensing and crowdsourcing applications primarily focus on performing tasks wherein the work can be assessed and validated by other peer workers and experts. For example, tasks such as sensing public phenomena and reporting locally relevant information are relatively easy to be verified with multiple workers by assigning them the same task. However, when it comes to collecting individuals' personal contextual and

activity data, it is much more challenging to assign peer workers to verify the data collected by a worker. After all, it would be infeasible to assign peer workers to follow and observe data collectors recording his or her daily activities. Perhaps because of this challenge, we have seen a lack of study evaluating the effectiveness of mobile crowdsourcing for collecting individual activity data, including investigating participants' behavior in collecting the data. However, we argue that this gap needs to be filled because of an increasing need of collecting contextual and activity data in real word settings.

In our own study, we addressed this assessment challenging by asking participants to wear a wearable camera to capture their outdoor travel activities. Then, we used passively logged location and activity traces on their smartphones along with the photos captured by the wearable camera to reconstruct their travel activity histories during the study. Although combination of the three sources was laborious, it enhanced the validity and the reliability of the travel activity histories, which enabled us to use them as a ground truth for evaluating participants' collected travel activity data when comparing among different approaches.

5.2.2 Acquiring Annotations on Recorded Activity Data

Researchers in context and activity recognition routinely collect labeled contextual and activity data for building training, and testing their systems. While it is impossible to conduct a comprehensive review of this line of research, we rather focus on the research that particularly aims at supporting acquiring annotations. One focus of obtaining annotations is to leverage video to help recognizing collected activities. For example, CRAFT (Nazneen et al., 2012) adopts both *in situ* and *post hoc* approaches to capture behaviors of children in a video. However, in their study, p*ost hoc* annotations were added by experts to validate *in situ* annotations added by parents. The annotators were not people who

performed the activities. In addition, the study was not aimed at comparing performances of different approaches in the field.

Another topic relating to annotation acquisition is reducing the effort required to provide annotations. One approach is asking users to speak to annotate (Harada et al., 2008; Lane, Xu, et al., 2011). Another is using a Context-Triggered ESM prompt to ask users to label activities (Cleland et al., 2013). (Cleland et al., 2014) compared the accuracy of labels using this approach with using both structured and semi-structured approaches where researchers annotate the activities. They found that the accuracy of labels obtained using the Context-Triggered *in situ* approach was similar to the structured approach. However, in this study only the Context-Triggered ESM approach was not conducted in a controlled setting. In addition, the authors neither analyzed the quantity and quality of *recordings* nor analyzed users' experience with or behavior while using the approaches. To the best of our knowledge, our study is the first study providing a systematic analysis of different approaches to collecting annotated activity data and investigating participants' behavioral of collecting activity data in the field. We also further provide thorough suggestions on the approach, tool and instruction for using mobile crowdsourcing to collect activity data.

5.2.3 Validity Assessment of Research Methods

Another research area related to this study is assessing the validity of approaches to collect behavioral data. In this line of research, methods often being assessed are usually retrospective methods such as surveys (Sonnenberg, Riediger, Wrzus, & Wagner, 2012) and interviews (Klumb & Baltes, 1999) because they are generally believed to be subject to recall errors. To validate data collected via these methods, researchers have used ESM or Ecological Momentary Assessment (EMA) as a "gold standard" to compare with retrospective methods because ESM and EMA are considered to accurately reflect participants' *in situ* experiences and

behaviors. In addition, the daily construction method (DRM) (Kahneman, Krueger, Schkade, Schwarz, & Stone, 2004), an approach proposed for allowing participants to reconstruct the sequence of activities that occur during a day, has also been assessed using ESM/EMA (Dockray et al., 2010; J. Kim, Kikuchi, & Yamamoto, 2013). However, data collection for context-aware systems development introduces new concerns that go beyond validity as compared to a gold standard, for example, the quantity and the temporal alignment of collected activity data compared to the actual activity.

5.2.4 Mobile Receptivity and Interruptibility

Finally, finding opportune moments to request users to perform data collection tasks is critical for maximizing response rate. This topic has received attention from a number of researchers, including those employing an ESM approach for issuing requests to obtain data for developing machine learning models (Turner et al., 2015). When using an ESM to prompt users to respond to annotation task (e.g. a questionnaire), one question is: how receptive are users to an annotation task on mobile phones? Research on receptivity has focused on developing models for predicting users' interuptibility (Rosenthal et al., 2011), attentiveness to communication (Dingler & Pielot, 2015; Pielot, de Oliveira, et al., 2014a), availability for calls (Pielot, 2014) and boredom (Pielot, Dingler, San Pedro, & Oliver, 2015).

On the other hand, recent research also investigates users' attentiveness to mobile notifications. Overall, the research suggests that mobile users are quite attentive to mobile notifications. For example, Alireza et al. (Sahami Shirazi et al., 2014a) suggested that mobile users valued notifications related to people and events more highly than otherwise. Both Pielot, et al., (Pielot, Church, et al., 2014) and Chang & Tang (Y.-J. Chang & Tang, 2015) found that mobile users attend to notifications typically within several minutes; and Chang & Tang (2015b) further

suggested that mobile users are more likely to attend to messages within a minute when their phone is not silent than when their phone is silent. In addition, Dingler and Pielot (Dingler & Pielot, 2015) found that mobile users were attentive to messages 12.1 hours a day, and they would return to their attentive state within 5 minutes after inattentiveness.

Recent research also explores opportune moments to deliver notifications to mobile phones. For example, Fischer et al. (J. E. Fischer et al., 2011a) suggested that at the endings of making calls and receiving SMS indicated breakpoints on the use mobile phones. Poppinga, et al. (B. Poppinga et al., 2014) developed a model for predicting opportune moments to deliver notifications. They suggested that phone position, time in a day, and location were good indicators of opportune moments. Pejovic, et al, (Pejovic & Musolesi, 2014b) explored opportune moments for delivering questionnaires and suggested that good indicators of opportune moments included physical activity, location, time of day, and engagement. Finally, Sarker, et al. (Sarker et al., 2014) found that location, emotion, physical activity, time of day, and day of the week played an important role in predicting availability for answering an ESM questionnaire.

However, while these research works suggested that mobile users are attentive to mobile notifications and indicated several features indicative of users' receptivity to messages and questionnaires, none of these research was addressing mobile users' receptivity to data collection tasks, especially when the task involves users annotating the activity. As (Turner et al., 2015) point out, most works in this line of research has focused on particular scenarios, making the applicability of the features predictive to people's receptivity to other scenarios uncertain. In particular, the scenario studied here—collecting and annotating activity data on the go—has not been studied.

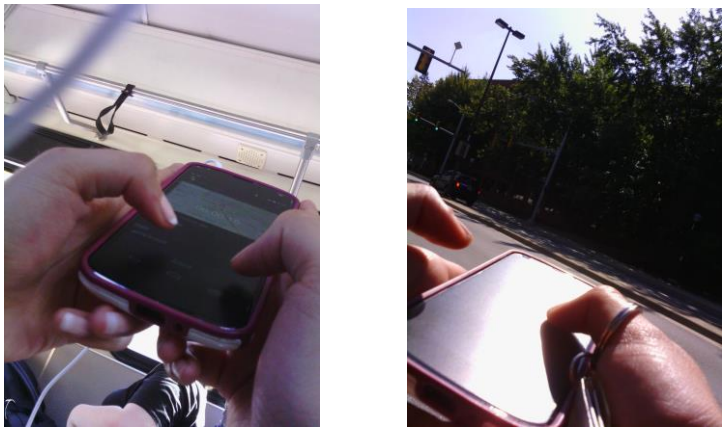


Figure 5.1 Study participants recorded and annotated their trips when they traveled outdoors.

As a result, we include receptivity analysis in our study, and our results suggest that, in the context of receiving prompts for annotating individual activity data, mobile users' receptivity was significantly lower when the users were in an activity requiring their high attention (e.g. driving) than in an activity requiring low attention (riding as a passenger). We believe this finding is important to using ESM for delivering mobile crowdsourcing tasks, especially to mobile users who are on the go.

In Section 3 below, we present our field study investigating the mobile crowd using three different data collection approaches to collect activity data in the field.

5.3 The Field Study

5.3.1 Collecting Travel Activity

We chose *travel activity* as the target activity to record and annotate. We had considered other types of contextual/activity data collected in prior research, including home activity, phone placement, noise, and body motions. We set up a list of criteria to evaluate each choice, including: 1) the data collection task is challenging enough but not too difficult so that users' performances could be

distinguished; 2) the task could be performed for several days, so that there is diversity within the to-be-recorded activity; 3) a known method exists for approximately detecting the to-be-recorded activity with a reasonable accuracy so that we could use it for implementing Context-Triggered approaches and 4) the occurrence of the to-be-recorded activity should be frequent enough so that failing to detect an instance of it will not lead to significant user frustration and a delay of the study. After evaluating each alternative, we chose to collect travel activity: *participants recording and annotating their trips when they are traveling outdoors,* as shown in Figure 5.1.

5.3.2 Choices of Approach to Compare: PART, SITU, POST

We chose to compare three approaches to collect transportation activity data, which are: Participatory Sensing (PART), Context-Triggered In Situ (SITU), and Context-Triggered Post Hoc (POST). We chose these three approaches for several reasons. First, PART and POST are commonly adopted and discussed techniques in mobile crowdsensing (Ganti et al., 2011; Khan et al., 2013; Lane et al., 2010). SITU implements a Context-Triggered ESM approach, which is commonly used for collecting contextual and behavioral data (e.g. (Froehlich et al., 2007c). Second, PART, POST, and SITU impose different kinds and levels of effort on users, namely, 1) the effort of operating the system to record and to annotate data; 2) the effort of remembering to start and stop recording data, 3) the effort of responding to a prompt in time and then returning to the original task if the current task is interrupted, and 4) the effort of recalling and reconstructing what happened during the recorded activity. We assume the differences in these aspects would influence user burden and compliance, and the quality of the recorded data. Finally, all PART, SITU, and POST have been used in collecting transportation data with users' inputs (Auld, Williams, Mohammadian, & Nelson, 2009; Froehlich et al., 2009; S. Reddy et al., 2010). Later we will describe the implementation of the three approaches in our study.

5.3.3 Instrument for Data Collection: Minuku

For this study we used *Minuku* to collect data. Minuku is an Android data collection tool developed in our lab and is supported by a backend for data storage. It can passively record contextual data (e.g. location, activity), trigger actions such as delivering questionnaires based on the context, and schedule daily diary prompts at designated times. These features are necessary for SITU and POST: Minuku needs to automatically record data when it detects that a user is likely traveling using a particular transportation mode (TM). In addition, in SITU, Minuku needs to prompt the user to annotate their trips when it infers the TM of the user. Minuku utilizes the Google Activity Recognition service[17] to generate activity logs, which are in turn used to generate a first approximation of users' TM. Specifically, Minuku extracts the "in vehicle," "on foot," "on bicycle," and "still" labels from the service, and uses a finite state machine to determine whether a user is in a certain TM or is stationary. Determining a start and an end of a TM requires consistently receiving the same activity labels in a window of time (e.g. one minute). We iteratively tested different window sizes for different TMs with some *ad hoc* experimentation until the TM detection was robust and accurate in our own testing and in a field pilot study. The testing and the pilot study were important to the experiment because while a low threshold would cause Minuku to repeatedly prompt users during the same trip (over-segmentation), a high threshold would impose a significant delay before Minuku detects the start of a trip.

---

[17]https://developer.android.com/reference/com/google/android/gms/location/ActivityRecognitionApi.html

5.3.4 Study Design and Procedure

We adopted a within-subjects design for this study, i.e., each participant collected data using each method: PART, SITU, and POST. We chose this design because we anticipated that people would have varied number of trips in a day and different commute routines. To mitigate the order effect, we randomly assigned participants to one of the six possible orderings of the three approaches. The number of participants in each order was balanced.

5.3.4.1 Collecting Trips using Minuku when Traveling Outdoors:

We asked participants to record and annotate their trips when they were traveling outdoors (i.e., between locations). The annotation interface was same for all three conditions and is shown in Figure 2a. Participants were asked to choose an activity type (i.e. transportation mode) best describing their trip, and add a note to describe their trips. Specifically, we told participants, *"The note field is optional. However, it would be great if you could let us know what the trip is about, especially when the trip is atypical, such as you are stuck in a traffic jam."* One intent of this instruction was to *encourage*, rather than *require* the participants to describe their trips to reduce their burden. Additionally, participants were given the freedom and flexibility in typing a note so that we could explore the types of information participants thought would be relevant to travel activities. Furthermore, we also told participants that when a trip was being recorded, an ongoing notification icon would reside in the notification bar of the phone, and they could access the annotation interface by choosing that notification, as long as they saw that notification.

Participants were asked to record and annotate at least two trips per day. We clarified to them that a trip should contain a clear origin and a destination, and they did not need to record outdoor movement shorter than three minutes nor

indoor movement. At the end of each day, we tracked the number of recordings that participants annotated, and transitioned them to the next study condition once they had aggregated four days of annotated trips in the current condition. When the transition occurred, we sent them a new version of Minuku customized for the next condition. We told them that the four days of recordings did not need to be consecutive, and they should travel as they would normally do. We provided them with $24 for completing the three conditions. Participants were also rewarded 25 cents for recording each extra trip beyond the two required daily trips, and they could earn up to $10 for the extra trips.

5.3.4.2 Performing PART, SITU, AND POST

For the PART condition, participants manually started and stopped recording their trips using the interface shown in Figure 5.2b. They were instructed: *"Hit the Start button when you start your trip; hit the Stop button when you end your trip"*. They could also pause and resume a recording. Clicking the "Add Details" button brought them to the annotation interface. We told them that they could modify labels and notes for their trips in the Recording Tab, in which they could also see all recordings. In addition, we instructed them how to handle transitions between trips with examples and not to intentionally split a trip in the same TM into multiple recordings. We also clarified that whenever they switched to a different TM (e.g., walking after parking a car), they were starting a new trip.

For the SITU condition, we told participants that Minuku automatically detected their TM and would prompt them a phone notification to annotate their current trip as soon as a trip using a new TM was detected (as shown in Figure 2c). We told them that choosing that notification took them to the same annotation interface and that notifications were automatically dismissed when they were detected as having ended the current trip. We emphasized that they could only

annotate during the trip because there was no recording tab in this condition, but they should annotate while they were in a safe situation (e.g. not while driving).

For the POST condition, we told participants that Minuku automatically detected their TM but would not prompt them to annotate during a trip. Instead, any trips they completed would appear in the Recordings Tab (as shown in Figure 2d), and Minuku would remind them every day at 9 pm to annotate. This approach is similar to a daily diary study and the day reconstruction method (DRM) used for reflecting on life experience (Kahneman et al., 2004). The method has also been termed *prompted recall survey* in transportation research (Auld et al., 2009). We told participants that they could annotate their trips in the Recordings Tab at any time.

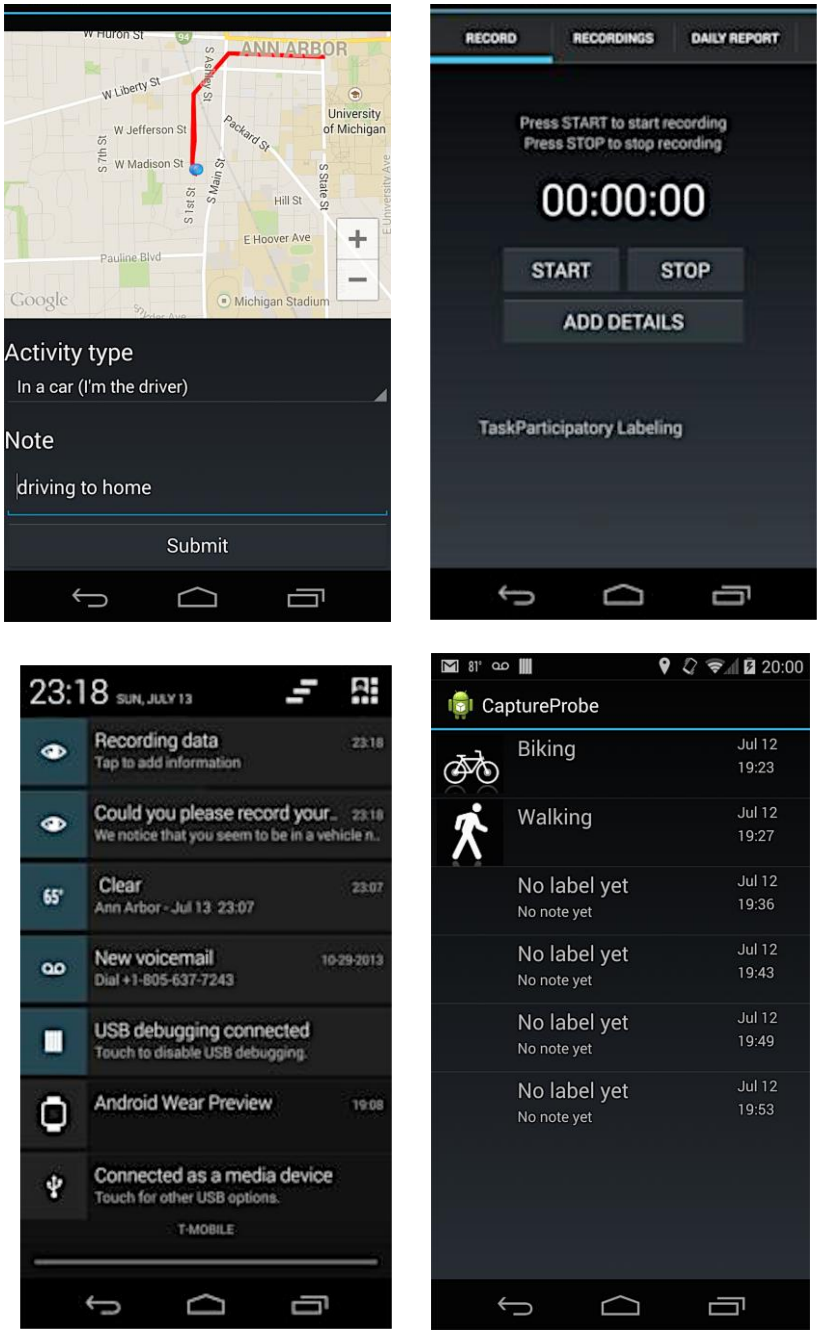


Figure 5.2 a) The interface for labeling and adding notes (left top), (b) PART: users manually record their trips (right top), (c) SITU: prompting users to annotate their trips (left bottom), (d) POST: users reviewing and annotating trips afterwards (right buttom)

5.3.4.3 Collecting "Ground Truth" Data

To assess amount and quality of a participant's recordings, it is necessary to know when he or she starts and stops moving outdoors. Therefore, we used Minuku to passively log participants' location and activity traces. While activity traces were passively logged at all times, location traces were logged only when participants were detected to be moving (i.e. not stationary) to minimize the power consumption of the phone. However, because location and activity traces are not always reliable and accurate, we asked participants to wear a wearable camera called Narrative Clip[18] during the study period. The camera is "always on" and takes a photo every 30 seconds. It is intended to be attached to the front of one's clothing, and to capture whatever the wearer is looking at. Wearable cameras have previously been used to validate travel diaries in transportation research (Doherty, Kelly, & Foster, 2013; Kelly et al., 2014). Inspired by the research, we intended to combine photos and logs to cross-validate and to generate *Ground Truth Trips* for each participant during the study.

We had considered recording continuous video, however, during the study, there was no wearable camera that could continuously record video for an entire day or take still photos at a rate higher than 2Hz. We asked participants to wear the camera at all times if possible, and emphasized to them that it was important for the study that they wore it whenever they started to move. However, for ethical reasons, we told them that they could take off the camera if they were uncomfortable with wearing it in particular settings. We told participants that photos were important for the analysis, but we did not tell them that photos were used as the ground truth. In addition, Minuku logged participants' actions related

[18] http://getnarrative.com/

to recording and annotation on Minuku and the times when Context-Triggered annotation prompts were generated.

5.3.5 Daily Diary and Post-Study Interview

In all three conditions, we sent participants a *diary prompt* e-mail at 9:30 pm daily to have them reflect on unlabeled recordings. The diary prompt contained a list of recordings captured that day, with the start time, end time, and a transportation mode label next to it. We asked them to review and correct any incorrect recordings. For any unlabeled recording, we asked them to choose a reason from a list of reasons why the recording was unlabeled and also provide context about the recording. We also asked them to list trips that they took but did not appear in the recording list, and to choose a reason for why the trip did not appear. We interviewed each participant after they completed all three conditions. We first asked them about their commute process in a typical day and how they decided which trips to record. Subsequent questions were focused on, for each approach, how they annotated, the challenges they encountered, their subjective preferences, and their suggested improvements.

5.3.6 Participants

We recruited participants that regularly commute to work or school by posting flyers on campus, sending department-wide e-mails, and advertising on social media. Respondents completed a screening survey to provide their 1) commute behaviors, 2) experience in using an Android phone; and 3) anticipated out-of-town travel plans in the near future. We filtered out participants who traveled fewer than 4-5 days in a week, whose typical commute time was less than 5 minutes, and who were planning to travel out of town for more than a couple of days during the study timeframe. We attempted to balance gender, age, and primary commute transportation mode among participants. While we started the study with 37 participants, only 29 completed participation (16 males, 13

females). There were several reasons why participants dropped out of the study: the app did not work with their phone, they lost the camera, or they stopped responding. Fourteen participants' ages were 18-25; twelve were 26-35, three two were 36-45, and one was over 55. We refer to them as P1-P29 throughout this paper. P13 and P19's data were excluded from the quantitative analysis because their data were incomplete. Thirteen participants reported that their primary commute mode was "car," while ten reported "bus," four "walk", and two "bike."

5.4 Data Processing and Coding

5.4.1 Cleaning, Merging, and Processing Recordings

It is important to distinguish between the terms *recording* and *trip* to correctly interpret the results of the study. In this paper, *recordings* refer to "recordings of trips" generated in Minuku using any of the data collection approaches, and *trips* refer to actual trips participants took during the study. That is, when Minuku records a trip, either via a context trigger or via manual activation by a participant, it generates a recording of the trip. As a result, it is important to note that a recording, though presumably representing a travel activity, does not necessarily perfectly reflect the actual travel activity. It is possible that a recording only captures a part of the activity or contains data beyond the activity (referred to below as *noise*). It is also possible that Minuku generates multiple recordings for one trip because the system stops and restarts recording during the same trip, either caused by the system or by the participant.

We collected in total 3070 recordings generated by Minuku. We firstly removed duplicate recordings generated due to Minuku's error. Then we inspected participants' diary entries to look for recordings explicitly mentioned by participants as errors or split trips. We removed false recordings and merged the mentioned split recordings. Through this data cleaning and merging process, we

obtained 2587 *valid recordings* (84.3% of all recordings), including both labeled and non-labeled ones.

5.4.2 Generating Ground Truth Trips

We reconstructed *Ground Truth Trips* from approximately 117,000 captured photos and activity and location traces. Several participants mentioned in the interview that they did not wear the camera at work or private places. There were also a few diary entries where participants said they forgot to wear the camera during a few trips. Thus, while we asked participants to wear the camera at all times if possible, we could not assert that Group Truth Trips captured "all" participants' trips during the study.

Two coders independently coded participants' Ground Truth Trip times from photos and trace logs. Coders were trained to infer a TM and when a participant started and ended a trip from photos. They were also trained to inspect activity traces using Google Earth for Desktop[19] to playback location traces to observe the movement of the participant. From these two processes, the coders then determined the final coded start and end times of each Ground Truth Trip. A standardized coding protocol was developed for the coders to follow to ensure consistency. One of the authors also met with the coders weekly to discuss and resolve any uncertainty on coded times. We randomly chose a subset (644) from the coder's' coded times and ran the intra-class coefficient (ICC) test between them. The ICC score was 0.87, indicating high reliability between two coders. After the test, each coder then coded a subset of the rest of the photos and logs (randomly assigned). We generated 1,414 Ground Truth Trips and paired each of them with participants' recordings by comparing start time, end time, and TM. Note that mislabeled recordings, recordings that were incorrectly labeled, were

[19] http://www.google.com/earth/explore/products/desktop.html

treated same as unlabeled recordings in the comparison. Thus, their corresponding Ground Truth Trips, if any, were also not counted as correctly labeled trips.

5.4.3 Analyzing Data in Two Phases

Because of the number and variety of data sources, we conducted two phases of data analysis. The first phase of analysis (referred to as Phase One) was primarily focused on the *comparison of the annotation approaches*, including comparing the quantity and quality of the recordings collected through each approach, as well as users' preferences of and experiences in using each approach. The results of the Phase One analysis have been previously reported in Chang et al. (2015).

The second phase of analysis (referred to as Phase Two) was focused on *user behaviors* while using PART and SITU in the field, which adds several new contributions to the Phase One analysis. The analysis includes a behavioral log analysis, a content analysis of participants' annotations, and qualitative analysis on participants' diary data. Additionally, we revisited interview data with a new theme focused on participants' overall strategies, behaviors, and challenges they encountered in using each approach.

In Section 5, we first present the analysis, results, and discussion in Phase One. Then in Section 6, we follow the same structure to report the new findings and offer new insights into users' behaviors of recording and annotation in the field obtained in Phase Two.

5.5 Phase One: comparing the annotation approaches

In the Phase One analysis, we first compare the quantity and quality of recordings obtained in each condition to the Ground Truth Trips we reconstructed. For comparing quantity, we measured the coverage of recordings. For comparing

quality, we measured completeness and precision of recordings. In addition, we measured overall performance such as number of recordings, recording labeling ratio, and recording annotating ratio.

5.5.1 Measures in Quantitative Analysis

5.5.1.1 Overall Performance Measures

We also computed measures that indicate participants' overall performance in producing different kinds of recordings using each method:

1. *Number of valid recordings*
2. *Recording labeling ratio*: The ratio of valid labeled recordings to total valid recordings
3. *Recording annotating ratio*: The ratio of annotated valid recordings to total valid recordings

5.5.1.2 Coverage & Trip Labeling Ratio

Coverage of recordings measures the length of data being recorded and *correctly labeled* in **absolute time** (seconds) and **percentage of total time** (percentage) per day. For example, if a participant traveled 70 minutes in a day and recorded 56 minutes, the coverage length is 56 minutes, and the percentage is 80%. The higher these two measures are for a particular approach, the greater quantity of data we collected through that approach. Another measure we calculated was *trip-labeling ratio (T-LR)* per day. This measure indicates the ratio of participants' actual trips recorded and labeled to total trips per day. For example, if a participant took 8 Ground Truth Trips in a day but only provided labeled recordings for 4, the T-LR would be 50% for that day. We hypothesized that T-LR of PART is lower than of SITU and POST because, in PART, participants had to initiate recording on their own, whereas in SITU and POST Minuku records a trip whenever it recognizes movement in a targeted transportation mode.

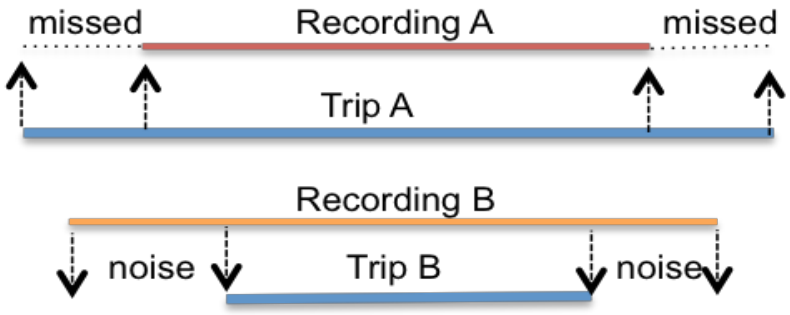


Figure 5.3 Noise and miss portion of recordings.

5.5.1.3 Completeness

Completeness measures the **percentage of a trip** being recorded and annotated. For example, if 15 minutes out of a 20-minute trip is recorded and annotated, the completeness of the recording is 75%. Two other related measures are the length of missed portions at the beginning and the end of a trip (seconds), respectively. If a recording starts ten seconds *after* a trip starts, it misses ten seconds at the beginning; if it ends ten seconds *before* a trip ends, it misses ten seconds at the end. We expected to see missed portion in recordings of SITU and POST because in both conditions Minuku needs to detect movement of the participant, which is likely to cause a delay in starting a recording.

5.5.1.4 Precision

Precision measures **percentage of a recording,** precisely reflecting its label, i.e. the activity. If a recording labeled as "driving" starts one minute earlier than the start of a 9-minute trip, it contains one minute of noise at the beginning, and its precision is 90%. We also measure the length of noise at the beginning and the end (seconds). Due to the detection delay, we expect to see some noise at the end of recordings of SITU and POST. An illustration of *completeness* and *precision* is shown in Fig 5.3.

5.5.2 Methods of Data Analysis

We used a Chi-Square Test to examine whether participants had significant differences in overall performance in producing recordings across three approaches. For measures related to coverage, completeness, and precision, we examined the main effect of variables of interest, including *condition*, *transportation mode*, *day of a week*, and *user* using an analysis of variance (ANOVA). The user variable was included to account for individual differences. We included the *periods of day* variable for trip level analysis such as completeness and precision. The periods we used are: morning (6am-11am), noon (11am-2pm), afternoon (2pm-6pm), evening (6pm – 9pm), night (9pm-1am), and midnight (1am-6am). These periods were determined based on our knowledge of participants' typical daily travel patterns obtained from the interviews. We also included the interaction effect between condition and transportation mode to examine whether certain combinations between the two would have an impact on recording coverage and accuracy. For example, in SITU, we expect participants to be less likely to label their trip when driving. We used the Tukey HSD Test for post-ANOVA pairwise comparisons.

For qualitative analysis, we transcribed interviews, and coded the transcriptions and daily diary entries using an iterative process of generating, refining, and probing emergent themes. The coding themes were focused on the topics of participants' likes and dislikes about each approach and their preferences and challenges of using the approaches.

5.5.3 Results: Quantity and Quality of Activity Data

5.5.3.1 Overall Performance

We start presenting measures of overall performance. Among the 2587 valid recordings, 1919 (74.2%) were labeled (i.e., were assigned a transportation mode), and 994 (38.4%) were annotated (i.e. contained a free-text note). As expected, the number of labeled recordings of PART (424) is noticeably lower than of SITU (723) and POST (772). In terms of the ratio of labeled recordings to total recordings, from highest to lowest are: PART (91.6%), POST (76.8%), and SITU (64.9%), and all of the differences between any two approaches are statistically significant using the Chi-Square Test for pairwise comparisons (PART vs. SITU: $\chi^2 = 109.9$, $p < .001$; SITU vs. POST: $\chi^2 = 33.4$, $p < .001$; PART vs. POST: $\chi^2 = 40$, $p <. 001$). This suggests that participants less often labeled recordings using the Context-Triggered approaches, whereas when they were able to record a trip in PART, they were very likely to also label it. In addition, PART also had the highest ratio of annotations to recordings (58.2%), which is statistically significantly higher than of SITU (31.6%, $\chi^2 = 25.1$, $p < .001$) and of POST (36.8%, $\chi^2 = 28.3$, $p <. 001$). No significant difference was found between SITU and POST. This suggests that participants also were mostly likely to write a note about a recording when they used the PART approach.

There are several things to note regarding these results. First of all, the SITU approach, i.e. asking users to label during traveling, led to the lowest ratio of labeled recordings. We think this may be linked to the issue of interruption in SITU. It was also likely that participants missed the prompt often. Secondly, the ratio of annotated recordings for POST is roughly as low as SITU. We speculate that this is because, in a post hoc review, it might be easier for participants to recall (or reason) the transportation mode of the trip than to recall details of the trip, making them less likely to annotate those recordings. Third, SITU and POST

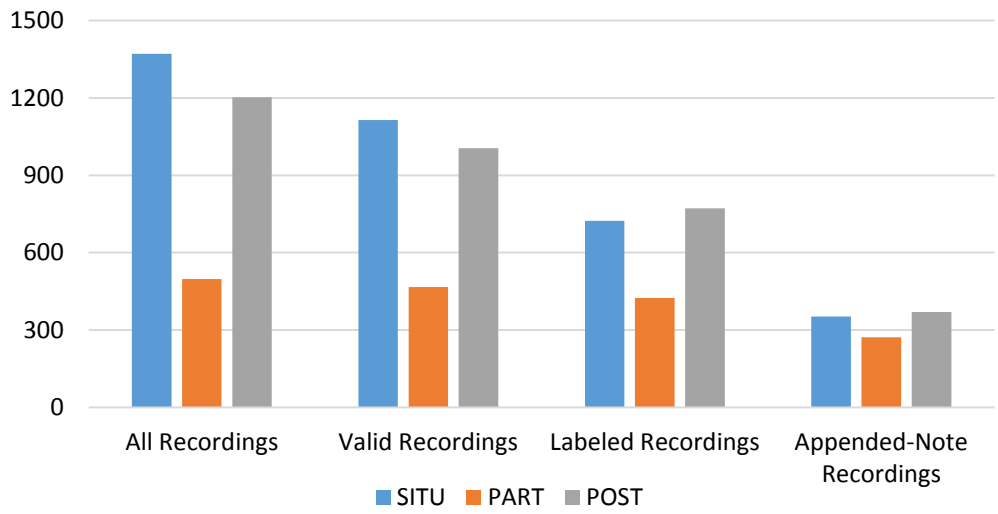


Figure 5.4 The differences in number of recordings decreased when as users' effort increased.

produced more valid recordings than PART because they employ automated recording. However, we learned in the interviews that participants sometimes were asked to label one trip more than once in SITU and POST because Minuku sometimes falsely detected them stopping and starting a new trip. Regardless of the reasons, Figure 4 shows that as the level of user effort increased (i.e. labeling and giving a note), the advantage of Context-Triggered approaches was diminished with respect to producing a larger number of annotated recordings. .The decrease in the rate of adding notes is especially apparent, possibly because we only *encouraged* instead of *required* participants to add notes to recordings, making this action more dispensable than the other requested actions.

5.5.3.2 Coverage of Recordings

In this section, we show that more labeled recordings, however, does not necessarily indicate a greater quantity of annotated activity data. We compared the ratio of actual trips being labeled to the total number of actual trips per day as

well as the coverage of recordings among the three approaches. Here, our results indicated main effects of transportation mode ($F[5,454]=5.3$, $p < .001$) but not condition. In a post hoc analysis, we found the ratio of recorded to actual walking trips to be lower than bus trips ($p < .001$) and car trips ($p = .02$), respectively. We think this may have been because participants considered car and bus trips more like "real trips," and may have been more likely to record and label them.

For coverage length, our results showed main effects of both condition ($F[2,454] = 4.9$, $p = .007$) and transportation mode ($F[6,454] = 18.6$, $p < .001$). In a post hoc analysis, unexpectedly, we found that the total coverage (absolute time) of PART is greater than that of SITU ($p = .02$) and POST ($p = .02$). A similar result was also found in coverage percentage: both condition ($F[2,454] = 12.9$, $p < .001$) and transportation mode ($F[5,454] = 2.8$, $p = .02$) had a main effect on coverage percentage. The coverage percentage of PART was greater than that of SITU ($p < .001$) and of POST ($p < .001$).

We found these results interesting and surprising. Although participants were not more likely to label more trips using any approach in a day, they produced a larger quantity of annotated travel activity data (in terms of length of time) using PART than using SITU and POST. Based on our observation of the characteristics of recordings, we conjecture that this might be because many of the recordings generated in SITU and POST were fragmented while the recordings generated in PART were more complete and precise. To confirm this hypothesis, we further analyzed completeness and precision of recordings.

5.5.3.3 Completeness of Recordings

As a reminder, *completeness* denotes the percentage of a trip that was recorded and labeled. If 15 minutes out of a 20-minute trip is recorded and annotated, the

completeness of the recording is 75%. Our results showed main effects of both condition ($F[2,1365] = 35.2$, $p < .001$) and transportation mode ($F[5,1365] = 8.2$, $p < .001$). A post hoc analysis showed that completeness of recordings of PART (68.2%) was significantly higher than that of SITU (48.1%, $p < .001$) and POST (47.4%, $p < .001$), as shown in Figure 5.5 This result supports our hypothesis that recordings in PART were more complete than the recordings in SITU and POST.

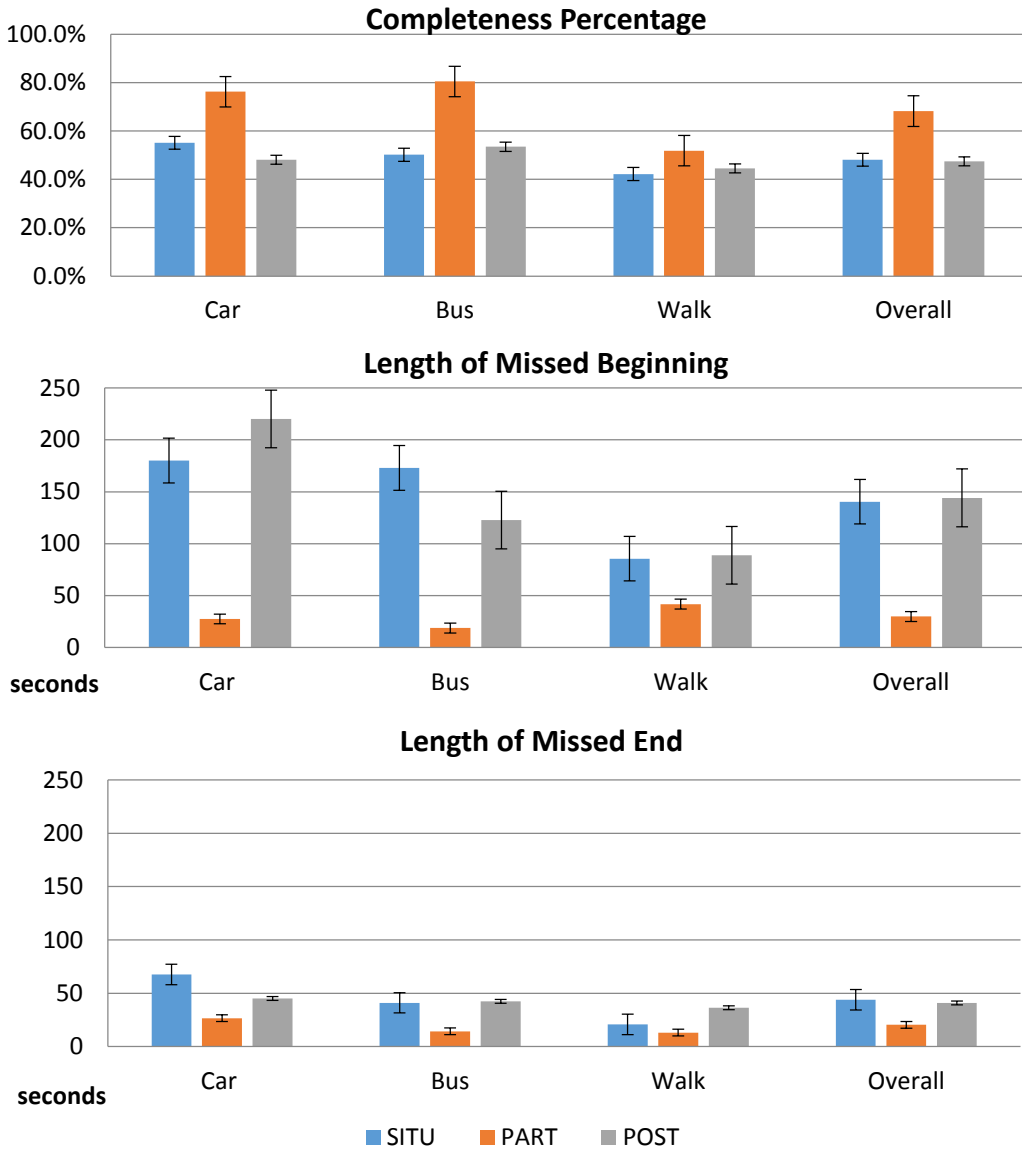


Figure 5.5 Completeness of Recordings (Left), length of missed portion at the beginning (Middle), and length of missed portion at the end (Right) across approaches and transportation modes.

We also found completeness of recordings for walking trips (45.2%) lower than of car trips (59.8%, $p < .001$) and bus trips (59.7%, $p < .001$). There also existed an interaction effect between condition and transportation mode ($F[4,1365] = 3.8$,

p = .004). In particular, we found that when using PART, completeness of recordings of walking trips (51.8%) was significantly lower than of bus trips (80.5%, $p < .001$) and car trips (76.2%, $p < .001$), respectively. This result may indicate that there was a larger disagreement regarding when walking trips started and ended between participants and our coders than car trips and bus trips.

We further looked into what led to the incompleteness of recordings. Regarding missed portions at the beginning of a trip, we found main effects of condition ($F[2,901] = 31.3$, $p < .001$) and transportation mode ($F[4,901] = 7.2$, $p < .001$), and an interaction effect between condition and transportation mode ($F[4,901] = 3.9$, $p = .004$). Specifically, recordings of PART missed significantly shorter portions at the beginning (29.8 seconds) than of SITU (140.4 seconds, $p < .001$) and POST (144.1 seconds, $p < .001$), suggesting that the delay of transportation detection did lead to longer missed portions at the beginning. In addition, recordings of walking trips missed longer portions at the beginning than of car trips ($p < .001$). We think this missed portion may be mainly responsible for the lower completeness of recordings of walking trips. On the other hand, there was no statistically significant difference in the length of missed portions at the end, among the approaches. The missed portions across the three approaches were also quite short. This result is not surprising because we expected that Context-Triggered approaches would tend to stop recording after the end of the trip because of the detection delay. On the other hand, this result also suggests that when using PART, if participants stopped recording before the end of the trip, they did not stop recording too early, which thus limited the length of missed portion. However, it should be noted that this result does not imply that participants stopped recording before the end of the trip. The completeness analysis only looked at recordings that had missed portions. We present precision analysis in the next section, which shows an overall measurement of how much noise was contained in participants' recordings.

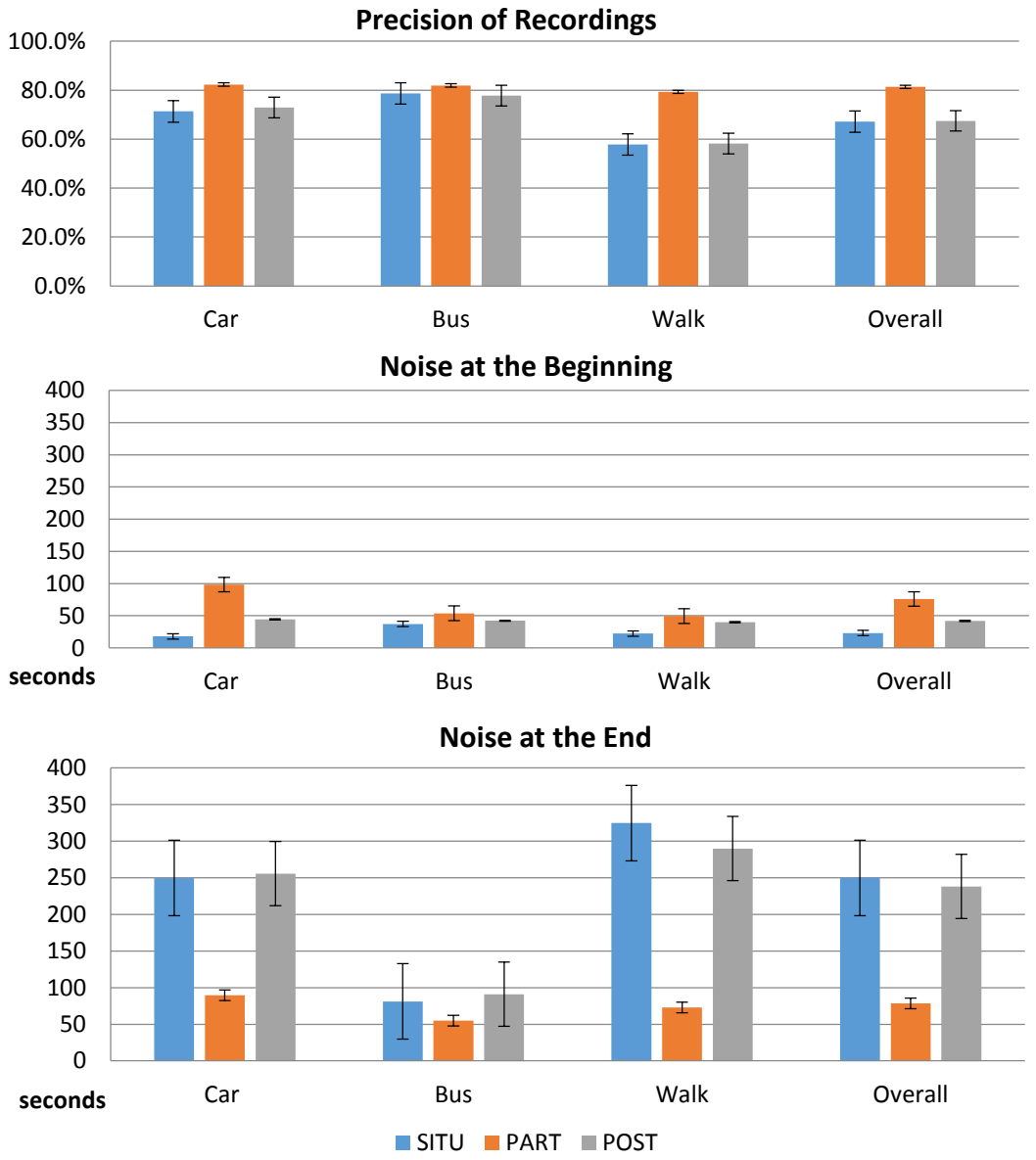


Figure 5.6 Precision of Recordings (Left), noise at the beginning (Middle), and Missed Portion at the end (Right) across approaches and transportation modes.

5.5.3.4 Precision of Recordings

As a reminder, *precision* measures the percentage of a recording reflecting its transportation mode label. If a 10-minute recording labeled as "driving" starts one minute earlier than the start of an 8-minute trip and ends one minute after the trip

ends, it contains one minute of noise at the beginning and at the end, respectively. Thus, its precision is 80%, meaning only the 8 minutes of the recording in the middle precisely reflect the label "driving." Our results showed main effects of condition ($F[2,901] = 32.1$, $p < .001$) and transportation mode ($F[4,901] = 16.5$, $p < .001$). We found the precision of recordings of PART to be higher than of SITU ($p < .001$) and POST ($p < .001$), as shown in Figure 5.6. As with the completeness result discussed above, this difference was probably caused by the detection delay. Furthermore, we found that the precision of recordings of walking trips in both SITU and POST was lower than any other combination of transportation mode and condition (all p-values are below .001). With further investigation, we found that the low precisions of recordings of walking trips of SITU and POST were mainly caused by the noise at the end, as shown in Figure 5.6 (bottom). Specifically, our results showed not only that recordings of SITU and POST contained significantly more noise at the end than of PART (both p-values <.001), but also that both recordings of car trips ($p = .005$) and walking trips ($p < .001$) contained significantly more noise at the end than did bus trips. We think these results may be because the ends of car trips and walk trips are more ambiguous than bus trips *in terms of TM detection*.

To summarize, our quantitative analysis indicates three results of particular interest. First, although SITU and POST produced more labeled travel activity "recordings", PART produced a greater quantity of annotated travel activity data in terms of coverage length of time. Second, recordings of PART were more complete (less missed data at the beginning) and more precise (less noise at the end) than recordings of SITU and POST. Third, it seems that walking trips are most ambiguous among the all TMs regarding when a trip starts and ends. However, it is important to note that these results did not suggest any tendency of participants' behavior in terms of whether they tended to record before or after the activity because these analyses did not combine the measures of completeness and

precision together. Instead, these analyses separate these two measures and primarily focused on contrasts among the approaches. We will dig more into participants' behavior in recording and annotation in the Phase Two analysis.

5.5.4 Results: Experiences in Using PART, SITU, and POST

5.5.4.1 Challenges Encountered

According to participants, the greatest challenge of using PART was to remember to record a trip. Most participants reported that they had forgotten recording their trips once or more. Furthermore, many participants reported that it was easier to forget to start recording than forget to stop because once they had started a recording, they were aware that Minuku was recording and would remember to stop it. Some participants also mentioned they took off the camera while they went indoors, and this action reminded them to stop the recording.

The greatest challenge of using SITU was being able to annotate during an activity before the prompt disappears, when the activity requires high attention. For instance, whereas most participants said it was not troublesome to annotate while walking, participants who commute by car reported that when driving they had to find a good time to label when getting prompted, usually at stoplights. In order not to miss the prompt, several participants said they tended to wait for the prompt once they started moving, but this gave them pressure and anxiety. For example, P5 said: *"it made me so anxious, like 'I've got to record this.'"* She continued: *"...at first I thought 'oh, [SITU] sounds like the easiest one' but it was actually annoying. [...] there was no way to go back and redo it afterwards, [so the] pressure was like 'I've got to record while I'm doing it or I'll miss it.'"*

The most-cited challenge of using POST was being unable to recognize a trip. While sometimes it was because when reviewing the trip on the map the

trajectory did not make sense to them, at other times they said they simply could not recall what a trip was about. For example, P26 said: *"[…] I did not recall anything, but it recorded itself. But at the end of the day I had to remember as to what I did at that point, what I did not do at that point."* Interestingly, when reviewing a trip, whereas some participants said that they relied on the map to recognize a trip, others said they mainly relied on the time of a trip. When asked about their rationale, participants who mainly relied on the time indicated that their schedule and travel pattern were regular and predictable; thus time was sufficient for them to recognize their trips. On the other hand, participants who often had irregular travels tended to rely on the map view to recognize their trips. However, participants generally agreed both maps and time were useful, and noted they had used both for labeling at some point during the study. It is noteworthy that participants often "reasoned" a trip rather than recalling it. For example, P22 reasoned her trips largely based on the time, *"I definitely looked at the times a lot because I know I'm walking between 4:30 and 4:45, and then I know I'm driving between 4:45 and 5:00 something, and then if I knew it was an evening trip, I'd remember if I drove or someone else drove." P24, on the other hand, used trajectories to reason her trips: "[…] like when the line is clearly on the bus route that I take, [it] is very obvious, so that's very reliable, and the same for a car and walking."*

5.5.4.2 Likes and Dislikes

Most participants liked PART because they had complete control over what and when to record. For example, P18 said, *"I guess the good part about participatory is that I wouldn't have to respond to three-minute walking trips 'cause those seemed not important."* In addition, they thought the PART approach produced the most accurate recordings among the three. Participants disliked PART mostly because they had to remember to start and stop on their own. For example, P5

said, *"You had to remember to press. […] so if you were forgetful you wouldn't want to have that burden."*

Participants disliked SITU mainly for being prompted erroneously—sometimes multiple times during a single trip. For example, P10 complained about getting prompts whenever he encountered a stop sign: *"By the time I get to the stop sign, it was [like]: 'Perfect, you got a stop sign.' And then, [the prompt] would then pop up. I was like, "You stupid [app], [Do] not give me the notification."* Another commonly cited problem was being unable to prevent the app from recording the movement they did not want to record. For example, P13 said, *"[…] especially when I didn't wanna record a trip, it would constantly be nagging me. Like when I work, I deliver stuff."* P5 also complained: *"... it would record me walking inside, […]. I was like 'ugh, just leave me alone.'"* Furthermore, participants felt that they lacked control over when to annotate in SITU, as P24 reported: *"I didn't like how I couldn't go back to my trips at the end of the day. Like I said, every now and again, I was concerned about not being able to record them... I couldn't go back and see which ones I forgot to record."* These participants wished there had been a way for them to review and labeled their trips afterwards like in POST.

On the other hand, participants liked SITU for its prompting feature suggesting the current transportation mode. For example, P4 said, *"I like that it did have that reminder, it was able to pre-judge what transportation I was actually using"* P9 also said, *"[…] it was pretty efficient the way that it only prompted when it was a long trip."* He later added, *"I thought [it] was intelligent. It can detect when you're in a car, when you're walking, so, which was pretty good. […] It was always accurate."*

Participants liked POST in that they only needed to annotate their trips once at the end of the day or when they were free, as P34 said, *"I really enjoyed being able to […] fill it all out in one time. […] It gave me a lot of flexibility. I could label it*

afterwards. I could label it at the very end of the day when I was sitting down charging the camera." When asked to rank the three approaches, P28 described an improved version of POST by saying, *"The best one would be: have an app which will do efficient tracking, and it will pop up only once in the night. It will do everything in the background, okay?"* However, not all participants liked repeatedly annotating their trips all at once, which may have led to less effort being directed towards the annotation task. For example, P29 illustrated this issue in the interview: *"Submit. Submit. Submit. [laughter]. Most people will be more diligent so they'll take more time to fill out the reports."*

Another often mentioned dislike about POST was seeing a number of errors, such as trips that were too short to record or trips that were hard to recognize. For example, P9 said: *"Prompting me for a lot of trips which weren't trips actually. [...]. I couldn't remember what they were, because the map would show like 10 feet or something, like a dot."*

5.5.5 Discussion of Findings in Phase One

5.5.5.1 The Pros and Cons of PART, SITU, and POST

We draw on the findings and discuss the pros and cons of PART, SITU, and POST in three aspects vital to collecting annotated activity data through the mobile crowd: *quantity of data*, *quality of data*, and *user experience*.

Quantity of Data

One question for a Participatory approach (PART) versus a Context-Triggered approach (SITU and POST) is: Does automated recording lead to a greater quantity of data compared to manual recording? Our results do not suggest such an advantage. Not surprisingly, the Context-Triggered approaches did generate a considerably larger number of recordings than PART. However, participants did

not necessarily label those recordings. In the SITU condition, in particular, participants reported that in many cases they missed the prompt or intentionally skipped the prompts. This suggests that while a Context-Triggered approach may capture many activity instances, participants do not necessarily respond to them (i.e. annotate the data).

Furthermore, despite the ability to generate more recordings and capture more activities, the Context-Triggered approaches, overall, did not result in a greater quantity of data in terms of length of time in our study. We found that it was because many recordings of SITU and POST were less complete and fragmented. In addition, some of these recordings were parts of the same travel activity; they were split because of the over-aggressive segmentation caused by the false transportation detection. Note that the TM detection of Minuku was developed on top of the Google Activity Recognition Service with improvements on accuracy. Nevertheless, false detections are still unavoidable when it is applied to different people's activity patterns and to a variety of real-word settings.

Finally, although we originally expected that participants in the PART condition would record fewer trips in a day than in the SITU and POST conditions because of the higher burden, our result show that there was no difference in the number of trips recorded and labeled per day across the approaches. On the other hand, the coverage of recordings and the ratio of labeled trips to total trips between SITU and POST are similar. This seems to indicate that neither the interruption issue of SITU nor the recall bias issue of POST lead to a smaller amount of correctly annotated activity data, as compared with the other.

Quality of Data

Our result shows a pattern regarding completeness and precision of recordings. Both the Context-Triggered approaches, SITU and POST, had more missed

portions at the beginning and contained longer noise at the end due to the detection delay. In particular, the Context-Triggered approaches, in our study, seemed to detect the end of walking and driving trips less accurately than detecting the end of bus trips. This caused significantly more noise at the end of walking and driving recordings. Moreover, because of occasional detection errors, both SITU and POST had issues with splitting a single trip into multiple recordings, meaning that many recordings were fragmented. If we had had a more accurate activity detection, the noise and the fragmentation issue might be ameliorated. However, it is likely that researchers who aim to collect activity data using a Context-Triggered approach may not yet have a full-fledged context detection system that is accurate or intelligent enough to prevent false detection and to accurately select what and when to record. As a result, researchers who attempt to use a Context-Triggered approach may need to expect these errors in the data.

In contrast, recordings of PART more precisely matched the actual start times and end times of their corresponding Ground Truth Trips. Moreover, participants also stated that they felt their recordings in PART more accurately reflected their actual travel activities. However, there were times, although not often, participants forgot to stop recording their trip because of some distraction or because they had been preoccupied with other matters, which resulted in a few recordings with noises at the end.

User Experience

According to the qualitative findings, we identify two key aspects of user experience particularly vital to collecting annotated activity data: *user burden* and *user control*. Regarding user burden, participants generally felt PART least convenient because they needed to remember to record their trips. In contrast, they appreciated the convenience of SITU and POST because of their automated

recording and prompt, especially that in POST, they did not need to annotate during the activity in the field as they needed for the PART and SITU approaches.

Regarding user control, participants highly valued being able to control when and what to annotate and record. The fact that participants could only annotate during a trip in SITU made participants anxious about missing a prompt, especially when an activity required their attention (e.g. driving). They favored the flexibility of deciding when to annotate in POST because they could annotate whenever they were free. In addition, participants wanted to control the instrument so that it did not record a trip they were reluctant or did not need to record. However, as mentioned earlier, these issues are specific to Context-Triggered approaches and can be challenging to address due to the lack of a full-fledged context detection system for employing this approach. On the other hand, we think these issues are crucial to address because inaccurate detection is likely to annoy users over time with recurring prompts and thus decrease users' compliance. One solution is allowing users to take control over the recording process when context-detection is not accurate. As context detection improves, users may be willing to cede more control to the system. To summarize, we think it is important that future mobile crowdsourcing tools take both user burden and user control into account to assure good users' experience in recording and annotating activity data. Neglecting either of these two aspects may result in a decrease of users' compliance. However, it is also noteworthy that these two aspects are in tension with each other because more control may lead to more burden. Future research would be needed to explore an ideal combination of the two aspects to make users' compliance more sustainable.

5.6 Phase Two: User Behavior Analysis

In the first phase of analysis, we focused on comparing the three approaches in terms of the activity data collected and the user experiences. In Phase Two, we

primarily focused on understanding *user behavior* in the field, i.e. how participants recorded and annotated their activity using PART and SITU. It is important to note that, in the previous phase, we observed some behavioral aspects of the participants. For example, participants generally were able to record their activity precisely using the PART approach and they seemed to miss longer portions at the beginning of walking trips, suggesting a delay in recording walking activity. However, in this section, we dig more deeply into participants' behaviors by inspecting their interactions with Minuku in using the two approaches. In Phase Two, we focus on *activity type* instead of *transportation mode* in terms of the impact o annotation behavior. As an example, instead of distinguishing between cars and bus, we distinguish between Drivers and Passengers. We make this distinction because we think these two activity types demand different degrees of attention from participants, which we think would be influential on when and how participants would annotate their travel activity when traveling. These activity types were provided by the participants' assigned labels to each trip.

In addition, it is important to note that we did not we analyze participants' behaviors in POST despite the fact that we did collect and organize the data in this condition. We chose only to focus on PART and SITU because we were mainly interested in understanding user behaviors "in the field," i.e. when participants were mobile and situated in a travel activity. Although participants sometimes annotated their recordings when they were on the go in the POST condition, most of our participants, according to the interviews and based on our preliminary inspection on the behavioral logs, more often annotated their recordings at the end of the day at home (usually after receiving the annotation reminder). Below, we provide more details of the analysis and the findings.

5.6.1 Behavior Log Analysis

As mentioned in Section 3.4.3, we collected participants' usage logs in Minuku, representing all actions that participants performed within the tool. Analyzing these logs allowed us to understand *when* and *how* participants recorded and annotated their travel activities. Specifically, we measured: a) when participants started and stopped recording in PART, b) when participants started, submitted, and completed annotations using both PART and SITU, and c) how many sessions (a series of actions performed in a continual manner) participants undertook to complete the entire annotation process. After obtaining these measures, we examined the influence of *activity type* on these measures, i.e. whether a difference in these measures existed among different travel activities. The activity type for each recorded trip was determined by the user's assigned label and was classified into three categories: *driving*, *riding as a passenger* (whether by bus or by car), and *walking*. These are common travel activities, yet they demand different degrees of attention from participants. As a result, we expect to observe some differences in participants' annotation timings during different travel activities. In addition, we also compared participants' recording times with the Ground Truth Trips to examine whether they tended to start/stop recording their trip earlier or later.

In analyzing participants' behaviors, we had different specific research questions for PART and SITU because of their different mechanisms for collecting annotated data. For PART, we analyzed the influence of activity type on users' annotation completion time—the elapsed time of users' last annotation submission in relation to the start of recording. That is, we aim to investigate whether users would tend to finish the task right after they started recording, during the trip, or after the trip. Because the elapsed time is highly correlated to the length of the trip, we classified the completion time into three levels of an ordinal measure: START (3)—completing annotation within 60 seconds after the

start of the recording; DURING (2) —completing annotation between one minute later the recording and before the end of recording; AFTER (1)—completing annotation after the end of the recording. The 60-second threshold was decided based on two observations: a) a typical duration that were sufficient for participants to type a note with enough details (e.g. *"'traffic was VERY heavy due to rush hour'"*), and b) the distribution of the annotation completion times in PART (participants' annotation submissions started to scatter throughout the recording after 60 seconds, shown in Fig3b in the next page). We made sure that the duration was long enough for participants to type details because we would examine the influence of annotation timing on the length of notes in a statistical analysis. We refer to this ordinal measure as *Annotation Completion Timing* in the rest of the paper. A higher rank indicates an earlier time for completing the annotation task.

For SITU, we investigated participants' receptivity to annotation task requests. For our purposes, an annotation task was "responded to" by a participant when the participant started to annotate through the prompt. Our measures included: a) the percentage of annotation prompts responded to by the participants, b) how quickly the participants responded to requests, and c) how quickly the participants completed the requested annotation tasks. These measures displayed how receptive participants were when they were requested by a researcher to collect annotated activity data. We did not measure participants' recording times in the receptivity analysis because Minuku automatically started recording on its own in the SITU condition when it prompted participants. We also grouped participants' annotation completion times into START and DURING using the same 60-second threshold (there was no AFTER for SITU).

Finally, we inspected participants' behavioral logs to look for emergent patterns that recurred and were distinct from participants' typical patterns in recording and

annotation. From this inspection, we were able to uncover issues causing erroneous activity data. We also measured the length of annotations and investigated the influence of activity type and annotation completion time on the length and content of notes.

We ran mixed-effects regression models for all of the quantitative analysis. Specifically, we ran mixed-effects linear regression on numeric dependent variables (e.g. recording time, annotation time, the length of note), mixed effects logistic regression on binary dependent variables (e.g. whether an annotation prompt is responded to), and mixed-effects ordinal logistic regression on ordinal dependent variables (*Annotation Completion Timing*). For all analysis but one we included Activity (Driver, Passenger, Walking), periods of the day, and day of the week as fixed-effect independent variables. We used transportation mode (car, bus, walk) rather than activity (Driving, Riding as Passenger, Walking) for the analysis of response rate because we did not know whether or not a participant was a driver or passenger if they did not respond to the annotation task. We could have inferred this information from the ground truth photos of the wearable cameras. However, this inference would be unreliable. When coding photos, we found that it was difficult to distinguish between driving and being a passenger when the camera was not facing toward to the front.

5.6.2 Qualitative Analysis

5.6.2.1 Content Analysis of Annotations

We conducted a content analysis of participants' annotations (i.e. notes that participants added to recordings). Note that users were given freedom as to whether to provide an annotation and what to write in the annotation. Surprisingly, even when the participants were aware that the annotation field was optional, they provided 272 annotations in the PART condition (64% of 424

labeled recordings) and 352 annotations in the SITU condition (49% of 723 labeled recordings) for SITU. Two co-authors of the paper independently coded the recorded annotations obtained in PART and SITU. The codes were categorized into various categories such as routes (departure, destination), the context of the trip, intent behind/purpose of the trip, routineness, and errors. We assessed the inter-rater reliability (IRR) of the codes and obtained a Cohen's kappa value of .90, which indicates a high agreement between the two coders on the coded content and characteristics of annotations.

5.6.2.2 Diary and Interviews

We also analyzed participants' diary entries and revisited the interview data with a new focus on user behavior. Specifically, for diary entries, we focused on reasons why participants did not record and/or annotate their travel activities. For interview data, we sought to understand participants' overall recording and annotation behaviors and strategies in using PART and SITU as well as the issues they encountered that might have interfered with their recordings and annotations.

5.6.3 Results: Recording and Annotation Behavior

5.6.3.1 Recording Timing in PART

Our first result is regarding recording timing. We want to examine, overall, whether our participants would tend to record before or after the start of a travel activity. When we compared participants' recordings in PART with Ground Truth Trips, we found that, on an average, participants started recording their trips 46.2 seconds earlier than the start of the trip (Median=28, SD=221), and stopped recording 58 seconds after the end of the trip (Median = 28, SD=218). In particular, 72.4% of recordings started earlier than Ground Truth Trips, and 78.5% of their recordings ended later than Ground Truth Trips. Furthermore, we

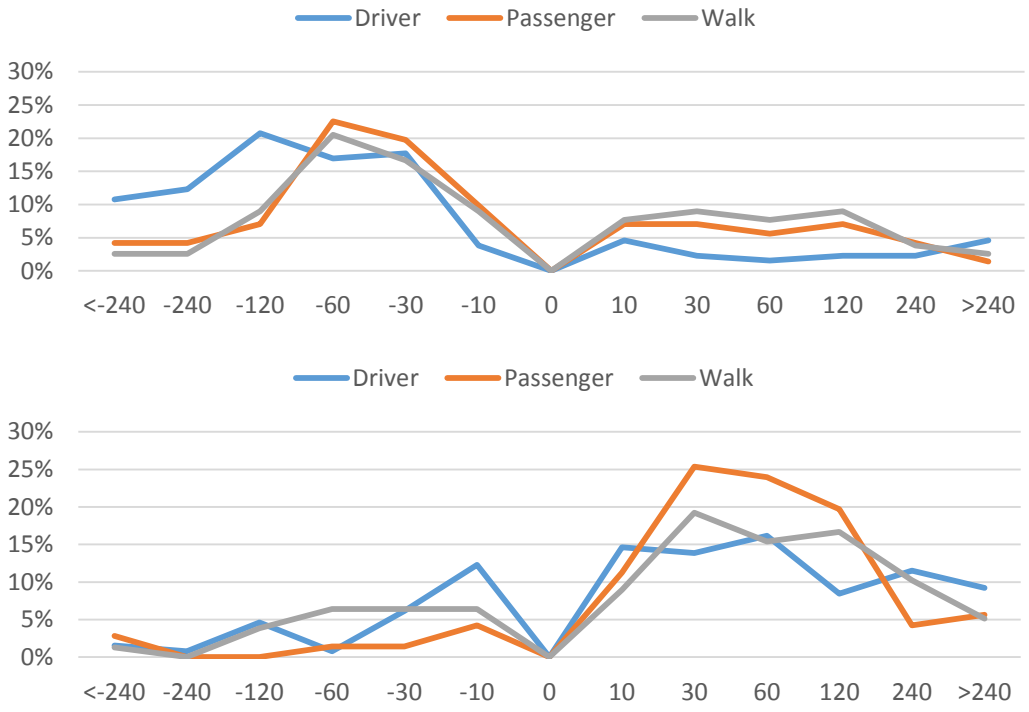


Figure 5.7 (top) Most recordings started before the actual trips, and Drivers started earlier than others. (bottom) The end times tended to occur after the end of trips, though there was no difference among activities regarding when late recordings were stopped.

found when participants were Drivers, they recorded their trips earlier than when they were Passengers or Walking (Figure. 5.7), and the difference between Drivers and Walking was statistically significant. (t (X) = 2.6, p=.01). We did not observe any statistically significant differences across activity types in terms of when recordings were stopped. These results complement well the results obtained in Phase One. That is, although in the PART condition participants produced recordings both with miss portions (i.e. record after the activity starts) and with noises (i.e. record before the activity starts), respectively, overall, participants more often recorded before rather than after the start of the activity, regardless of the activity type). The impact of activity type was mainly on *how*

much earlier participants started the recording. As a result, researchers may expect to see more often noise than miss portions at the beginning of recordings when the use a Participatory approach.

5.6.3.2 Annotation Completion Timing in PART

In the analysis of Annotation Completion Timing in the PART condition, we found a strong effect of activity type on Annotation Completion Timing. Specifically, we found that participants were more likely to complete annotation tasks at the start of the trip or during the trip when they were Passengers (M=2.51, SD=0.89) than when they were Drivers (M=1.70, SD=0.74, p=.01) and Walking (M=2.10, SD=0.93, p=.05). Figure. 5.8a shows when participants completed their annotation tasks using PART. When the participants were Passengers, 88.7% of annotation tasks were completed during recording; however, when they were Walking or Drivers, only 59.5% and 41.1% of annotation tasks were completed during recording, respectively. In other words, when participants were Drivers, nearly 60% of annotation tasks were completed after recording. This pattern is also supported by the number of sessions participants spent to complete annotations. Our results showed that when participants were Passengers, 94% of annotations were completed in one session; in contrast, only 60% and 64% of annotations were completed in one session when users were Drivers or Walking, respectively. The difference between Passengers and Drivers was statistically significant ($t(389) = 2.4$, $p=.02$), and between Passengers and Drivers was marginal ($t(389) = 1.78$, $p=.07$). On the other hand, we observed that among annotations completed during recording, participants tended to complete annotations sooner rather than later (as shown in Fig. 8b): a large part of annotation tasks were completed within one minute (Driver: 71%, Passenger: 77.5%, Walking: 77.3%). This suggests that when participants were able to complete annotations during recording, they tended to do them sooner than later.

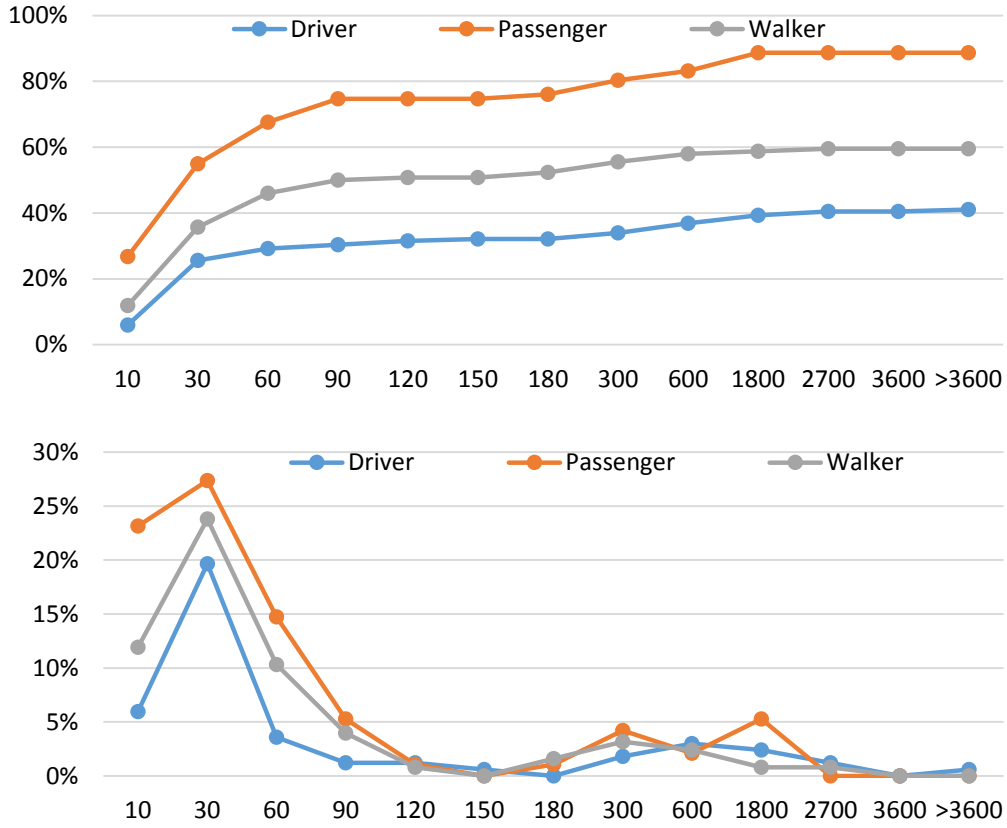


Figure 5.8 Annotation Completion Timing in PART. (a) Top: cumulative percentage of annotations completed during recording. (b) Bottom: the percentage of annotations completed between certain time during recording

In the interviews, we asked participants when they annotated their recordings. Many of them reported that they preferred to annotate soon so that they would not forget later. Participants especially mentioned that they would annotate soon when they were taking buses or walking because did not need to concentrate as they needed to when driving. For example, P9 reported: "*I'm sitting in a bus anyway, so there's nothing to do. You can just quickly do it if you're sitting in the bus. […] Walking also, there's nothing to do, right? You only have to walk.*" In contrast, when participants were Drivers, they needed to concentrate on driving;

they reported that when driving they annotated while they were at breakpoints (e.g. stoplight) or after they stopped their trips. As P26 said, *"If I'm walking I do it pretty [much] right away because it's not much of a deviation. If imagine I'm in a car, then I generally respond to it whenever I think it is safe or whenever I kind of stop the car."*

5.6.3.3 Users' Receptivity to Annotation Requests in SITU

In the analysis of receptivity, we examined participants' response rate, how quickly they responded to prompts, and how quickly they completed requested annotation tasks. For the analysis of "response rate" in particular, as noted earlier, we had to use transportation mode (Car, Bus, Walking) rather than activity type because we did not have reliable information about whether participants were Driver or Passenger in the recordings they did not label. We found that on an average, participants had high response rate to annotation prompts across all transportation modes (Car: 86.7%, Bus: 88.9%, Walking: 81.1%), and the effect of transportation mode on response rate was marginal (Car vs. Walking: $z(454)=1.9$, $p=.05$); Bus vs. Walking: $z(454)=1.6$, $p=.12$).

However, among notification prompts that were responded to, participants responded more quickly when they were Passengers and Walking than when they were Drivers (Passenger vs. Driver: ($t(418)=-3.79$ $p<.001$); Walking vs. Driver: ($t(418)=-3.31$ $p<.001$)), as shown in Fig. 9a. They also completed annotation tasks more quickly when they were Passengers ($t(417)=-2.8$ $p=.006$)) and Walking ($t(417)=-1.9$ $p=.06$)) than when they were Drivers (see Figure 5.9b). Interestingly, similar to the behavior in PART, we found that when participants had responded to a notification prompt, most of the time they submitted their annotation within a minute (Passenger: 95.7%, Walking: 94.7%, Driver: 89.45%). The differences between Passengers and Drivers and between Walking and Drivers are both marginally significant (Driver vs. Passenger: $t(417)=1.94$,

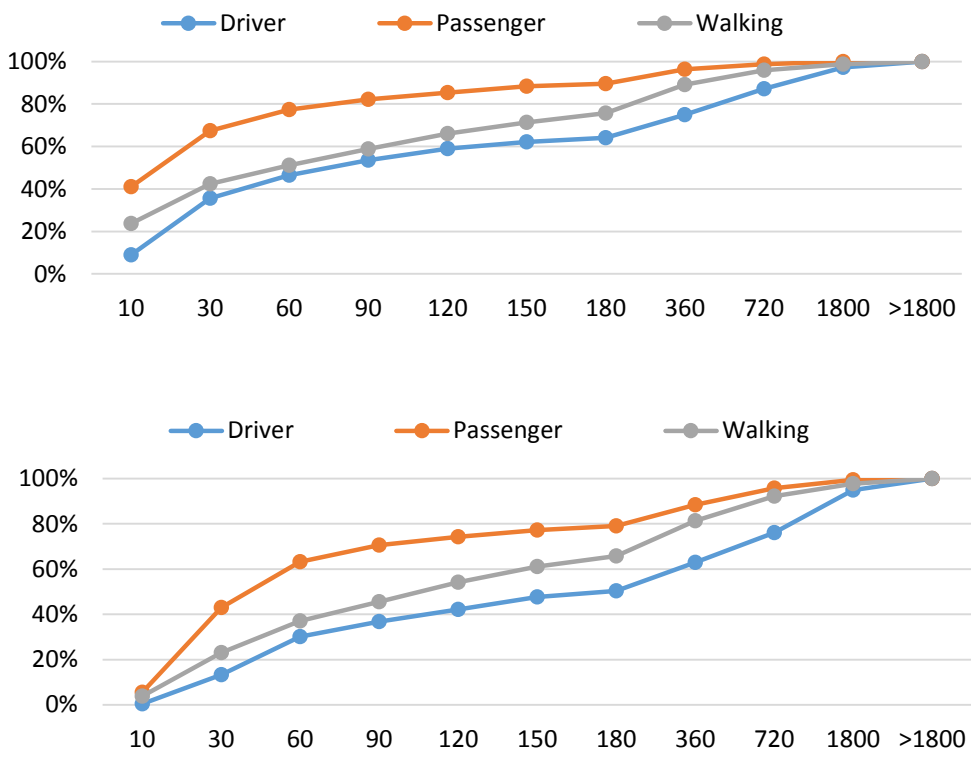


Figure 5.9 (a) Top: Cumulative percentages of annotation prompts responded to within certain time in SITU. (b) Bottom: Cumulative percentages of annotation tasks that were responded to and completed within certain time.

p=.05; Driver vs. Walking t(417) = 2.00, p=.05). These results seem to suggest that although activity type influenced how quickly participants could complete an annotation task after they have responded to it, its main impact on receptivity seems more related to how quickly participants could respond. In addition, regardless of whether participants were in the PART or SITU condition, they both tended to complete annotation sooner rather than later.

From the interviews, we asked participants about when they annotated their trips in SITU. Most participants were well aware that they were in the study and would expect to get prompts when they were traveling. A typical explanation for their

immediate response to the prompt is as what P4 said: *"I know it's gonna pop up sometime here soon. I just kept looking at my phone. I gotta remember that it's going to come up."* Many users added reasons why they preferred to annotate immediately. Similar to using PART, the main reason for performing it early was to prevent them from forgetting to do it later. P15 said: *"I immediately respond, so that I don't forget it later so, 'Okay. I've seen the notification so let me get over with it now.'"* P5 also said: *"I just want to get it done. I didn't want to miss it. [...]. I was very careful at the beginning and then I was worried, because I wanted to do it right away*."

On the other hand, when participants were driving, they deferred the submission to a point where they felt safe to complete it, as P22 said: *"I'd get the notification while I'd be in the process of driving so I'd have to wait 'til I was at a light or something, and kinda answer it or try to remember not to hit "Submit" [laughter] and then set it down, then go about my business."* However, sometimes it was hard for users to anticipate how long a breakpoint (e.g. stop light) is; thus, some users would defer it until the end of the trip before the notification disappeared. For example, U36 stated: *"In many cases that I don't know how much time I have at the light. And rather than, just leave it in the middle, I'd wait till I wasn't traveling anymore."* Taking these results together, participants seemed to prefer to respond to the prompt and complete the annotation task early if they are not preoccupied by the activity. We think this behavior cannot be all attributed to the fact that they could only annotate during the trip because participants also displayed the same tendency in using the PART approach, for which they could annotate whenever they wanted to.

5.6.3.4 *Characteristics of Participants' Annotations*

To understand how annotation timing and other factors would affect the characteristics of participants' annotations, we analyzed the effect of activity type on the length of and conducted a content analysis of all submitted annotations. For the former, we observed an interaction effect between activity type and Annotation Completion Timing on the length of participants' notes, as shown in Figure. 5.10. Specifically, we found that when participants annotated AFTER recording when they were Passengers and Drivers, they tended to put longer notes than when they were Walking (Passengers: $t(1046)=1.95$, $p=.05$; Driver: $t(1046)=2.37$, $p=.02$). In addition, when participants annotated During recording, they also put longer notes when they were Passengers than when they were Waking ($t(1046)=2.88$, $p=.004$). However, we did not observe an effect of activity type when participants annotated at the START of recording. Instead, while participants annotated at the START of recording during walking, their notes were generally short. These results suggest that participants seem to put short notes when they are walking, regardless of the annotation timing and that when they put notes early in the trip, they tended to put shorter notes. They were able put longer notes when they were Passengers, which we think might be because

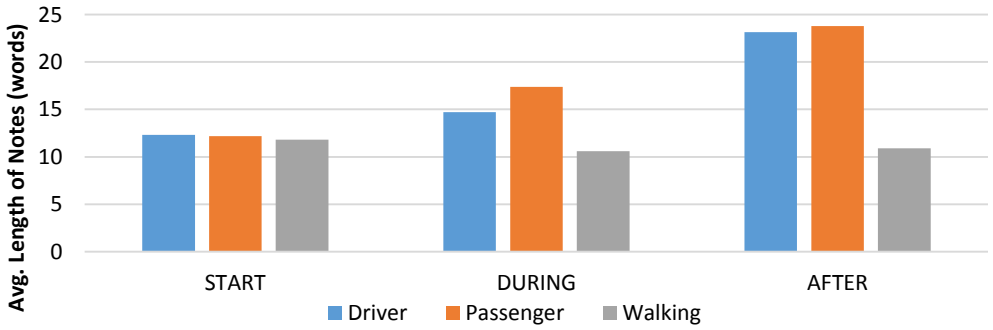


Figure 5.10 Participants generally wrote short notes when they annotated at the beginning. When users annotated AFTER recording when they were Passengers and Drivers, they tended to put longer notes than when they were Walking.

they have more attention available compared to other activity types. Finally, when they put notes after the activity, they tended to put longer notes, except when they were walking, perhaps because walking was too a routine activity for them. We investigated further into the content.

Regarding the content of annotations, we found different characteristics of participants' annotations according to transportation modes, activity types, and annotation timing. Specifically, we found some types of information appeared in annotations for one transportation mode more often than in those for other transportation modes. For example, participants more often described multiple-destinations (e.g. '*driving daughter to school then work*') and purpose of trips in annotations of car trips than in annotations of the bus and walking trips. We suspect that this might be because participants' bus and walking trips were more routine trips, whereas participants had car trips for more diverse purposes. We also found participants more often included information of transportation mode when they were walking (*e.g. "walking to the library where I volunteer twice a week"*) than when they were in a car or on the bus. Furthermore, when participants were Drivers and Walking, the annotations made at the Start contained fewer words and categories of information describing their trips. That is, whereas the annotations made at the Start mostly contained destinations and purposes of the trips, annotations made later included more details such as with whom the users were traveling, details of the route, and events occurring during the trip. However, when participants were Passengers, they use similar categories and number of words to describe their trips regardless of when the annotation was created. We think this might be because as a Passenger, participants had abundant time and cognitive resource to annotate during a travel activity. In contrast, when participants were Drivers or Walking, in which they had to spend more attention resources on performing the travel activity itself, participants did not seem to be able to include more information during the activity. Finally, we also found that

participants more often used shortened descriptions when they were Drivers, indicating their tendency of making notes as efficient as possible. Taken these results together, it seems that two major reasons are mainly responsible for explaining the characteristics of notes: what context is relevant to annotate about the activity, and the availability of participants for annotating the activity.

Another interesting observation we had is that some participants would assume a common ground shared with researchers, which made them shorten descriptions over time or referenced a trip to previous trips (e.g. *"to recycling from home"* ⇒ *"more recycling"* or *"walking to great clips to get haircut"* ⇒ *"'still walking to hair cut place"*). Some of these occurred because users were prompted multiple times in one trip in SITU.

5.6.3.5 Reasons for Unrecorded, Unlabeled, and Erroneous Activity Data

Finally, we analyzed diary entries, interview data, and inspected behavioral logs to identify reasons and patterns that caused unrecorded, unlabeled, and erroneous activity data. We found that forgetting and missed notifications were responsible for most of the unrecorded, unlabeled and erroneous activity data. Specifically, from diary entries, we found that the major reason contributing to unrecorded activity data was participants forgetting to record (18 out of 38 unrecorded trips). Other often cited reasons included feeling it was troublesome to record (8 out of 38) and feeling it was inconvenient to record (7 out of 38).

For unlabeled activity data, in PART the main reason reported by participants was forgetting to label (10 out of 23); in SITU, the main reasons were not part of their plan to annotate (93 out of 250) and missed notifications (88 out of 250). These results show that participants could either intentionally and unintentionally not

respond to a prompt, and it seems that both reasons, at least in our study, seemed to be of equal importance.

As to erroneous activity data, we learned from the interviews that many participants forgot to stop recording their trips after they had ended their trips when they used PART. This sometimes resulted in unnecessarily long recordings, large portions of which were incorrectly labeled. According to the participants, the main reason causing them to forget starting and stopping the recording was distractions in the moment or that they had been preoccupied with other things, as P22 said, *"So and then the one time I forgot to stop and transition from walking to driving… I just had a lot on my mind so I just didn't think about so I went all auto pilot."* U15 also reported, *"because you have to get down, you have to cross the street, you have to choose which shop to go to. So yes, I tend to forget here."* In particular, one common source of distractions reported was interacting with other people, as U20 said: *"Because oftentimes, I'd be wrapped up in what I'm supposed to be doing, or maybe I met a friend when I was walking, and we're walking together, and then I forgot."*

As to SITU, we observed from the behavior logs that many labeling errors occurred at transitions between travel activities. One typical case was that participants did not respond to the prompt until they were about to start a new trip. Another case was that participants changed labels because they thought they were transitioning to a new trip. For example, P10 commented his strategy of labeling his trip in SITU: *"If I got the notification right away when I was driving, then I'd put 'driving.' But since I would go back and I would always check it numerous times, so then over to walking instead of the driving, then I'd probably go and switch it to the walking."* In SITU, these issues seemed to be related to the delay of annotation prompts when participants transitioned to a new trip, and the issues often occurred when the transition was short such as walking to a car. While

participants were instructed to provide the transportation mode that was current as of when the prompt was issued, not when it was received, the participants would provide the mode when they responded to the prompt.

Below, we discuss the findings and conclude with implications for the design of a mobile crowdsourcing tool for collecting annotated activity data from individual mobile workers.

5.6.4 Discussion of Findings in Phase Two

5.6.4.1 Possible Reasons Behind the Influence of Activity Type

Our findings suggest that activity type influenced participants' recording and annotation timing, receptivity, and the characteristic of their annotations. Here we discuss the possible reasons for such influences.

First of all, regarding the recording timing, when participants were Drivers, they tended to record their trips earlier than when they are Passengers or Walking. We conjecture this might be related to the length of transition to the trip. For example, a transition to a car involves multiple stages (e.g. opening a door of the car, sitting in a car, and waiting for the car to move) and thus is longer than a transition to walking. As a result, participants would have more time to record at transitions to driving than at transitions to walking. Another reason that might explain the differences in the recording time would be participants' perception of the amount of attention required during the travel activity. Drivers might perceive a challenge of recording their trips precisely at the moment when they start traveling and thus tend to start recording earlier. On the other hand, although the transitions to bus trips might be longer than the transition to walking trips, the fact that being a Passenger requires limited attention to the travel activity might explain why the participants did not tend to record as early as for Driving.

The impact of the amount of attention required to perform an activity is also evident in the differences in the Annotation Completion Timing among different activity types. For instance, in both the PART and SITU conditions, although our results suggest that participants tended to annotate early rather than later, participants completed annotation tasks quickest when they were Passengers and slowest when they were Drivers. Moreover, participants also more often used multiple sessions to complete annotation tasks when they were Drivers and Walking. Drivers also completed annotations after recording in about half of the cases. In the SITU condition, participants also had a lower receptivity to annotation prompts when they were Walking and Drivers than when they were Passengers. These results taken together indicate that the level of attention required by an activity has an impact on user's Annotation Completion Timing. This observation was also supported by many participants' self-reports that they would annotate during breakpoints (e.g. stoplights) or after driving when they were a driver.

Finally, our results suggest that both Annotation Completion Timing and the context in which an activity is performed may have an impact on the content of annotations. For the former, annotations created at the START of a travel activity contained limited categories of the information about the activity than those created later. For the latter, participants more often described the purposes of their travel activities and included multiple destinations when they were in car trips than they were on a bus and walking. This difference might be because bus and walking trips participants recorded were more routine trips, whereas participants went to more diverse places when they were in cars. Finally, the context in which an activity is performed might also affect whether and to what extent participants were distracted or preoccupied. This might in turn influence how likely users would be to remember to stop recording.

5.6.4.2 Anticipating Characteristics of Collected Data

Following the discussion above, we summarize four features of an activity that may influence the quality and the characteristics of am activity recording and annotation. These features are a) length of transitions before and after the activity, b) degree of attention required for performing the activity, c) distribution and lengths of the breakpoints during the activity, d) the context in which the activity is performed. Specifically, based on our observations of the results, we first conjecture that recording timing mainly correlates to the lengths of transitions before and after an activity and the degree of attention required for performing the activity. The longer the transitions are, the more likely users may start recording earlier and stop recording later, respectively. Second, we conjecture that Annotation Completion Time mainly correlates to the degree of attention required and the distribution and the lengths of breakpoints during the activity. That is, the more attention is required for users to perform the activity and the fewer and shorter the breakpoints are, the more likely the users would annotate late or after the activity. Third, we conjecture that the content and the characteristics of annotations mainly correlate to the degree of attention required, the distribution and lengths of breakpoints, and the context in which the activity is performed. In other words, content and characteristics of annotations depend on not only how much time users can spend on annotation, but also what information is relevant to the current activity. The latter is especially true when the activity to be annotated is a routine activity in users' daily lives. The users may not only have a limited number of categories of information to describe the activity but also feel bored by annotating same information repeatedly. One example is that some participants shortened their annotations on the same and repeated activity.

Based on these conjectures, we present the mentioned features of activity in Figure 5.11. It is important to note that activity is a complex phenomenon (Nardi, 1996) and Figure 5.11 is only a provisional and simplistic representation of activity for the purpose of introducing the four features we found vital to collecting annotated activity data. We think this representation, however, may help researchers anticipate the characteristics of collected data such as how long the noise would be, how long a missed portion of an activity would be, what information would be included in annotation, and so on. For instance, if there tends to be a long transition to the activity of interest and the activity demands some attention from the user, researchers may anticipate that the user is likely to record before the activity starts and that the recording would contain some noise in the beginning. If the researcher anticipates that the activity of interest does not demand much continual attention or it does, but contains many breakpoints, the user may annotate early in the activity. One potential issue with early-made annotations is that the user may not mention events occurring later in the activity in the annotation unless they are explicitly instructed to do so. On the other hand, late-made annotations are also likely to neglect events that occurred early on, if additional salient events occurred later. Finally, researchers may be able to predict

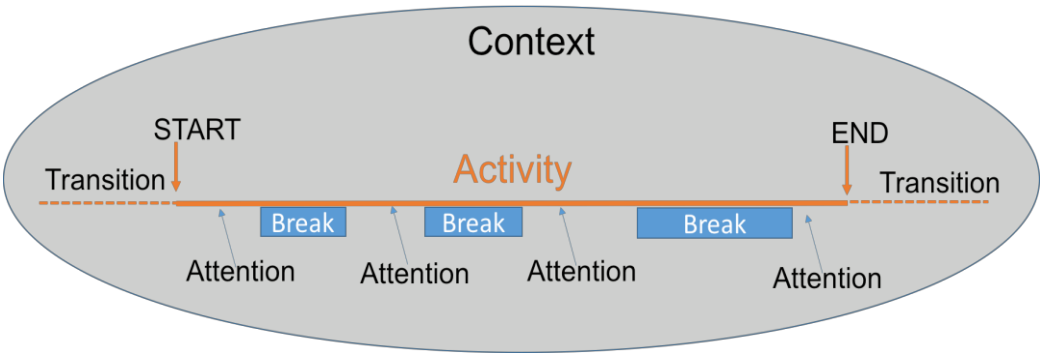


Figure 5.11 An activity with four features: a) length of transitions b) degree of attention required for performing the activity, c) distribution and lengths of breakpoints during the activity, and d) possible contexts in which the activity is performed.

whether users are likely to forget to start and stop recording using the PART approach by anticipating possible distractions during transitions before and after the activity. Researchers may also anticipate how likely the user may mislabel a previous activity using the SITU approach, given the length of transitions and the current context-detection method researchers use.

In the last section below, we conclude our findings with a list of suggestions for future work aiming to use mobile crowdsourcing to collect individual annotated activity data.

5.7 General Discussion

5.7.1 Towards a Better Practice of Collecting Annotated Activity Data

In this paper, we present a field study that aims to identify an approach that would be reliable and effective for collecting annotated activity data through the mobile crowd. Our study shows several important takeaways that shed lights on the approach, tool, and instruction that make mobile crowdsourcing promising for collecting annotated activity data. First of all, despite the fact that Context-Triggered approaches (POST and SITU) may produce a larger number of recordings, we show that many of these recordings may be fragmented and contain noise, resulting that Context-Triggered approaches may not necessarily produce a greater quantity of annotated activity data in terms of length of time. In addition, because of the presence of noise, data requesters would need to process and clean the collected recordings further to make them accurately represent users' labels. In contrast, despite the fact that the PART approach may produce a smaller number of recordings, the data produced by the PART approach are in general more complete and contain less noise. Regarding user experience, whereas the PART approach is generally more burdensome than both the SITU and POST approaches because it requires more users' effort, the SITU approach

can be sometimes considered interrupting or even annoying if it prompts users repeatedly or prompts them at situations when they do not want to annotate. However, a Context-Triggered approach is still worth the hassle to develop and employ because it reduces users' burden in operating the instrument. We believe that in the long run, reducing burden is a promising direction to pursue for sustaining users' compliance and participation.

Because of these tradeoffs between the PART approach and the Context Triggered approaches, we do not conclude that one approach is absolutely *better* than the others. Rather, we think a more important message of this paper is that we must understand the strengths and weaknesses of each approach so as to develop better practices for collecting annotated activity data via the mobile crowd. We also need to understand the tension between user control and burden and try to find a balance between them. While future research is needed to futher explore effective combination between thee two qualities, our tentative proposal is using a hybrid approach that granting flexible user control (e.g. PART) and use Context-Triggered as a support to reduce burden. We will be present details of this proposal in the next section.

Another important takeaway of the study is that we must understand users' behaviors in using each approach with respect to the nature of the activities being collected, so that we can better anticipate the characteristics of the collected data. We believe this understanding is crucial for knowing how to process the collected data and use them later for different purposes. It also informs how to improve the design of a data collection tool and how we should instruct participants to make a data collection process more effective. For example, our results show that participants tended to add annotations sooner rather than later and that the annotations created at the start tended to contain limited categories of information compared to those created later. While we cannot assert that these differences

should all be attributed to annotation timing, the fact that most of the time participants rarely revisited annotations made at the start implies that events occurring later in activities could be less likely to be mentioned in the annotation. These pieces of information (e.g. encountering traffic jam later in the activity), however, may be valuable for researchers to obtain so that they could use it for sorting out the collected data later or for inspiring what particular behaviors they would be interested in collecting more. In addition, when using the PART approach, participants tended to start recording before the start rather than after the start of a travel activity. As a result, researchers may expect to see noise at the beginning of the recording. When using the SITU approach, participants were more receptive to annotation tasks when they were performing an activity that demanded less attention (e.g. being a Passenger). Therefore, when researchers attempt to select whom in the field to request data collection tasks, it is vital to take users' current activity into consideration. After all, not noticing the prompt was reported as a major reason responsible for unlabeled recordings using the SITU approach. It is likely that participants were too preoccupied in performing the activity to notice the prompt. This finding is consonant with the idea that finding opportune moments to deliver data collection prompts might help improve participants' overall attentiveness to the prompt, which in turns affect responsiveness. As a result, an ideal data collection instrument should be able to estimate users' receptivity to data collection to reduce the likelihood of unlabeling because of their inattentiveness. Finally, another main reason for unlabled recording is "not part of the plan." This option was included because we assumed that researchers who request data would give participants freedom regarding which of their activities to record. This freedom is also wanted by the participants in our study. However, this option might not uncover the actual reason for which participants did not annotate a recording, such as whether participants were not available, or were not willing to annotate the recording. We believe in future

research it is important to distinguish between these two reasons to investigate how context relates to these two reasons respectively.

5.7.2 Design and Methodological Implications

Based on the takeaways mentioned above as well as other findings reported, we propose a list of design and methodological implications that aim to inform the approach, the tool, and the instruction for mobile crowdsourcing. Our goal for these implications is to improve the overall quantity and quality of the collected data as well as to sustain users' compliance. We combined implications for approach and tool in Section 7.2.1. Then in Section 7.2.2, we provide suggestions on instructions.

5.7.3 Suggestions for the Approach and Tool for Activity Data Collection

Our high-level suggestion on the approach and tool is to employ a hybrid approach, using the PART approach as the main approach to grant user control and use a Context-Triggered technique as a support to ease user burden, to remind users, and to prevent data collection errors. This may be considered a type of an *in situ* prompt that allows post hoc annotation—a combination of the SITU and the POST approach. A high level rationale behind this hybrid approach is that while granting user control and easing user burden can be seen as a design tradeoff, our experiences convince us that these two elements can be balanced to improve not only user experience but also the quantity and quality of data collected.

Specifically, we suggest researchers encourage users to manually record their activity to increase the accuracy of data as well as to provide user control; meanwhile, a Context-Triggered function, if available, can run as a fall-back to deliver reminders and to enable automation when it is necessary. Regardless of whether the Context-Triggered function is activated or not, the tool should allow

users to control when they want this function to be activated to prevent the tool from recording and prompting them when they do not want to be bothered.

The Context-Triggered function provides several important benefits. First, it can trigger reminders when it detects that the users have forgotten to start recording their activity. The tool then can remind them to annotate. Similarly, when it detects that the users have forgotten to stop recording an activity, it can automatically stop recording. Although this may result in some portions of activity not being recorded, it would help reduce unlabeled data and prevent a long period of noise at the end of the recording, thus making the data cleaner. In addition, the reminder notification should reside in the notification center even after the trip has ended. A reminder residing in the notification center during the activity will increase the users' awareness of an ongoing recording and allow them to annotate it at breakpoints. Leaving the reminder in the notification center after the activity ends provides the users with more control over when to annotate. It also avoids unnecessary pressure and anxiety of needing to complete an annotation task during the activity that demands high attention. The annotation reminder can indicate an aggregated number of recordings waiting for the users' responses. This may make the users mindful of the presence of unannotated recordings and remind them of annotating sooner while they still have a fresh memory of what happened during those activities.

To ameliorate the issue of mislabeled recordings, a Context-Triggered function can detect whether an activity to be annotated is likely to be a transition (e.g. a short walk to taking a bus). When detecting such an instance, a reminder can ask users to verify whether their label should be associated with the transition activity (walk) or the next activity (bus). Another alternative to avoid mislabeling errors is to let the instrument start recording only after the users have responded to the annotation prompt instead of at the moment of detecting the activity. This will

assure that the label provided by the users correctly reflects the activity being recorded at the moment when the users see the prompt. Finally, to further ease user burden, the Context-Triggered function can suggest a label where possible, meaning that the users only need to change the label if it is incorrect. When the tool detects the users being in the same activity consecutively, it asks whether this is a continued activity, and if yes, it automatically connects the current recording to the previous one. Detecting an opportune moment for delivering the prompt during or after an activity can also avoid interrupting the user.

5.7.4 Suggestions on the Instructions for Activity Data Collection

Regarding instructions, because users may tend to start recording before the activity and stop recording after the activity, we suggest that researchers explicitly instruct users to be as precise about the recording timing as possible to reduce noise in recordings. However, as it is not always convenient for the users to operate the tool at when the activity starts and ends (e.g. driving), the tool may allow researchers to enter anticipated lengths of noise at the beginning and the end of the recording, respectively, and trim the recording accordingly. In addition, because users may tend to annotate sooner rather than later in the activity and do not often revisit early-made annotations, we suggest that researchers instruct users to be mindful about the events occurring after they complete annotations and encourage them to revisit annotations after the activity. On the other hand, we also suggest researchers interested in knowing more about the semantics of the activity instruct users to include the intent behind or the purpose for the activity in the annotation, especially early in the activity because they will remember it better. From our experience in analyzing the content of the annotations, we found this information particularly helpful for understanding the meaning of an activity to participants. Since this information would be difficult to infer from the raw data, we believe it is worth instructing users to add it in the annotation when researchers think that help interpret the collected data. For example, although

intent information may not be essential for detecting the activity per se, it is useful for distinguishing among variances within the same activity, such as identifying personally significant places, predicting where the user is departing for, and recommending places of interest for travel activities (Andrienko, Andrienko, Mladenov, Mock, & Pölitz, 2010; Ashbrook & Starner, 2003; Baltrunas, Ludwig, Peer, & Ricci, 2011; Bhattacharya, Kulik, & Bailey, 2012; Cao, Cong, & Jensen, 2010; Liao, Fox, & Kautz, 2007).

With these improvements proposed, our future work includes both implementing these features on Minuku, the instrument we used for the study, to implement the hybrid approach, and to examine whether the proposed features would increase the effectiveness and improve the user experience of the process of collecting annotated travel activity data with the mobile crowd. We plan to employ the tool and the approach to collect other activity types. Meanwhile, we hope that these design suggestions will enable researchers and practitioners interested in using mobile crowdsourcing to collect activity data to collect a greater quantity and quality of activity data and annotations.

5.7.5 Limitations

It is important to note that the study is subject to several limitations. First, the Ground Truth Trips were reconstructed where photos were available. As a result, despite the fact that we instructed participants to wear a wearable camera for an entire day, we were not 100% sure whether they wore the camera all day. This might make the photos subject to a systematic bias related to the availability of photos. Second, the sampling rate of the camera is one photo per 30 seconds. Although we used logs to establish more precise times of Ground Truth Trips, there might be still some imprecision on the start/end times. Third, we do not know whether users were passengers or not in a car when they did not respond to

an annotation prompt. As a result, in the analysis of response rate we had to use the transportation mode information from Ground Truth Trips instead.
Fourth, our analysis was based on a relatively small sample of smartphone users in a particular area. Their behaviors thus are not representative to the general mobile user population, especially that there exist differences in the dynamics of travel activities in different geographic areas.

Fifth, the study participants only used each approach for four days. Their compliance was likely to change if the study had been longer. For example, participants might have been less compliant in using the PART approach if the study had been longer because it is more burdensome. Therefore, in the context of long term participation (e.g. users signing up for contributing their own behavioral data for several months), it is unclear whether the PART approach on average can still achieve a higher coverage of behavioral data compared to Context-Triggered approaches. Furthermore, it should be also noted that the context of this study was that participants continuously and constantly collected their behavioral data in a certain period of time. As a result, the results of the study are likely not applicable to the the context in which researchers request data collection tasks only occasionally, such as only when they need specific data while they could not collect by themselves. The latter context differs from the former in the sense that is that users may not anticipate receiving a data collection request as they would in the former context. This difference may then affect the receptivity to the task. In addition, users' perceived obligation for collecting requested data might also be different. That is, instead of perceiving themselves as "participating in a study" and thus feeling obligated to being cooperative and complying, in the latter context, users may perceive themselves "helping data requesters" obtain data and feel less obligated to collect the data. It is likely that they perceive themselves simply offering a service or selling their data instead of contributing to research. It is thus unclear to what extent these differences would

impact users' performance and compliance in collecting behavioral data. And we believe it would be necessary for future research to reexamine the effectiveness of different data collection approaches in the latter context.

Sixth, another question is how the results of the study are applicable to collecting other types of or more complex activities. It should be noted that in this study we asked our participants to distinguish among different travel activities and to record and annotate them separately. The activities collected in the study thus have two characteristics. First, the travel activities collected in this study have a clear starting point (starting moving from the departure) and ending point (stopping moving at the destination). Second, travel activities including driving, taking a bus, and taking a train, in most cases, are mutually exclusive (excepting that walking can happen in some travel activities such as walking in a train). As a result, participants could only record and annotate one distinct travel activity at a time, and there was little ambiguity as to which activity to record and annotate. However, not all of our daily life activities have a clear starting point and ending point (e.g. having a meeting), and many of them can be undertaken simultaneously and/or embedded in a higher level activity. Thus, for example, if researchers are interested in collecting activities that could be performed simultaneously (e.g. eating and watching a video at the same time) or high level activities in which a number of low level activities can be embedded (e.g. eating, watching a presentation, and discussing during `a "lunch meeting"), it may be ambiguous for users to know "which activity" or "which part" of an activity to record and to annotate about. Furthermore, a higher level and a complex activity can have a number of dimensions to be recorded and annotated about. Without specific and precise instructions of how and what to record and annotate, researchers may obtain varied contents in collected recordings and annotations from different users. Moreover, we should not assume that the researchers, developers, and designers, who desire to collect high level and complex activity

data always have a clear idea about which parts and dimensions to record and annotate about. As a result, we believe that the results of this study might be more applicable to collecting activities that have a clear and distinct category and that have a clear starting and ending points. As a result, we think future research is needed to examine the same research questions in the context of collecting other types and more complex activities.

Finally, the findings we presented in this study are largely tied to the instruction, the tool, and the approaches we used and evaluated to collect travel activity data in the study. The study results, including data quantity and quality, could have differed considerably if we had had a much more or a much less accurate transportation detection in Minuku. One large assumption of this study is also that a data collection tool is able to detect some kind of context or behavior from which a prompt for collecting the context or the behavior can be triggered. However, it is likely that developers and designers have a need for collecting the data before they have built a context and behavior detection function. In this case, the PART approach would be the only option for them. On the other hand, it is important to note that the goal of the study is to inform the tool and the approach for using mobile crowdsourcing to collect annotated context and behavioral data. As a result, we sought to understand the strengths and the weaknesses of each approach if they are all available for use. With this assumption, we believe the design and the instructional implications we draw from the findings do advance toward our goals.

5.8 Conclusions

In this Chapter, we presented a field study comparing three approaches involving the mobile crowd in recording and annotating their travel activities in the real-word setting. The approaches we compared are Participatory (PART), Context-Triggered in situ (SITU), and Context-Triggered post hoc (POST). To compare

the three approaches as well as to learn about how participants used the approaches to collect travel activity data, we adopted a mixed-method approach to collect and analyze various types of data from participants, including activity and location traces, photos from wearable cameras, behavioral logs on the phone, participants' recordings and annotations, daily diary entries, and interviews.

We conducted two analyses, each of which focused on different aspects of the data. In the first analysis, we focused on analyzing the pros and cons of the three approaches. We showed that although SITU and POST produced more travel activity recordings, PART produced a greater quantity of travel activity data in terms length of time. This suggests that automated recording was not advantageous in collecting travel activity in our study. Regarding data quality, recordings of PART were more complete and contained less noise than recordings of SITU and POST because many of the recordings of the latter were fragmented and contained more noise. In addition, we showed that participants highly valued being able to control what and when to record and annotate, and appreciated automated recording and reminders that could reduce their burden. As a result, we conclude that user burden and user control are two important aspects of user experience on the mobile activity data collection tool.

In the second analysis, we focused on investigating user behavior in the field, i.e. how participants used PART and SITU to collect data in the field. Our results suggest that the type of travel activity being collected influenced participants' recording timing, annotation timing, receptivity to annotation tasks when using the SITU approach, and the characteristics of annotations. In particular, participants tended to start recording before rather than after the travel activity. They also tended to annotate sooner rather than later during the travel activity when the activity being collected did not demand high level of attention resources. Finally, we presented reasons responsible for unrecorded, unlabeled,

and erroneous activity data. To respond to the findings, we have provided design and methodological implications aimed at making mobile crowdsourcing more user-friendly and more effective for collecting greater quantity and quality of activity data.

Chapter 6 Minuku: A Tool for Collecting Contextual and Behavioral Data

6.1 Introduction

Mobile phones have been a focus of researchers and practitioners for collecting behavioral and contextual data. The wide availability of mobile phones for mobile users, and the availability of various sensors and Internet access on mobile phones have appealed to a number of researchers for building mobile systems for collecting behavioral and contextual data. While sensor information allows researchers to make inferences of mobile phone users' activities at different times and places, the wide Internet access allows the phone to send collected data to a designated server almost everywhere, making it possible for researchers to track the progress as well as monitor the status of data collection. This has made mobile phones a great instrument for sensing public phenomena, known as a mobile crowdsensing (Ganti et al., 2011) and citizen science (Robson, 2012).

Moreover, since smartphones phones have been more affordable to consumers and an increasing variety of applications also have become available to consumers, smartphones have become not merely a communication tool, but also a personal digital assistant (PDA) and an informational and entertainment center. This transformation has moved many of the activities that are previously performed on a computer and physical objects (e.g. calendar, notebook) to smartphone *apps*, which means that more and richer contextual and daily behavioral data can be observed and captured on smartphones—application usage, personal schedule, notification attending behavior, social media usage, and more (Falaki et al., 2010; Rahmati & Zhong, 2013; C. Shin, Hong, & Dey, 2012; Q. Xu et al., 2011). These pieces of information can be further combined, aggregated, or sophisticatedly integrated for the purpose of understanding mobile users' behavior

and/or developing models for classifying or predicting mobile users' states and activities (Pielot, 2014; Pielot, de Oliveira, et al., 2014a; B. Poppinga et al., 2014; Smith & Dulay, 2014). With all of these becoming feasible, pragmatic, and promising to many fields of research, a variety of mobile sensing systems, frameworks, and libraries have been developed to support mobile data collection for particular applications (e.g. see Thebault-Spieker, 2012), as well as for generic use (e.g. Froehlich et al., 2007b). A number of tools are also developed for specifically supporting ESM, such as (e.g. Seo et al., 2011).

However, thus far, there has not been a mobile data collection tool gaining wide adoption. These tools are mainly leveraged by the research or development team that built the tools. Although various reasons can contribute to the relatively low adoption, I attribute it to three main main reasons: *configurability*, *flexibility*, and *timeliness*. First, configurability refers to two aspects: the extent to which a data collection tool can be configured for a particular project, and the ease of configuring the tool. Regarding the former, most of the previous tools aimed for a generic purpose provided some kind of configurability. However, because these tools require specific knowledge in a programming language to configure the tool, this requirement has hindered many researchers without the knowledge from using the tool for their own research. Second, even when a researcher has the knowledge for configuring the tool, configuration are not always flexible enough to fit the researcher's need. After all, what data to collect and when to collect participants' responses are largely dependent on the research questions of the study, which not merely vary across fields, but also vary across studies. Many research studies, such those adopting an ESM, focused on capturing data in very specific situations (context-based), at specific times, (schedule-based), or even for both (Capatu et al., 2014). To support behavior researchers to conduct different research studies, it is important to develop a research tool having great flexibility in configuration.

Finally, regarding timeliness, many previously developed tools have been incompatible with current mainstream smartphones because of the rapid advancement of smartphone systems. These tools are now functional on only a small set of phones due to the changes in platforms or to the system updates modifying several fundamental features of the prior systems. For example, in the recent five years, the Android mobile operating system has moved from 3.0 to 6.0, where the API changed from API 11 to API 23, a 12-level of difference[20]. Since 2012, Android has also had several significant leaps in the system API (4.2 to 4.3; 4.3 to 4.4; 4.4 to 5; and 5 to 6), each of which contained important library changes and has deprecated some previously commonly used functions to obtain contextual data. Without researchers' dedication in maintaining the tool, a research tool can soon become incompatible to new smartphones because of these system changes. This not only poses a great challenge on the researchers who strive to make these tools continuously available, but also makes researchers who are the users of these tools have difficulty knowing which research tool would work for the majority of current smartphones. Another timeliness related issue becoming increasingly relevant is whether the research tool can be extensible for obtaining data from external sources such as a wearable device, an virtual reality helmet, an external sensor, a data repository site, or an Internet of Things framework. Since these sources are increasingly available to researchers, we expect that more researchers would desire to leverage them to obtain more data about the context of the users. Thus, extensibility is a necessary capability a research tool ought to have to maintain its timeliness.

---

[20] https://en.wikipedia.org/wiki/Android_version_history

In this chapter, I introduce Minuku, an Android smartphone mobile data collection tool for addressing the three features aforementioned: configurability, flexibility, and extensibility. In addition to these features, I also aim to addresses some of the features proposed in Chapter 5 for supporting the collection of annotated behavior and contextual data. By focusing on these features, Miunku contributes to context-aware system development by enabling more researchers to collect annotated behavioral and contextual data with high quality and quantity. It also potentially contributes to mobile crowdsourcing/sensing, behavioral research, and any other research fields in which collecting behavioral and contextual data from smartphones is increasingly common. In the following sections, I will highlight the features of Minuku that make its capability and functionality beyond previous comparable tools, including a) the support of concurrent logging sessions, b) the support of monitoring customized *states* and *situations*, c) the support of situated actions and sophisticated scheduling, and d) its configurability, flexibility, and extensibility, and e) its support of different approaches for collecting annotated activity data. Then, I will present the implementation of the capability of Minuku and illustrate how they enable these features. Before we go into these details, below I first introduce the important concepts in the Minuku system.

6.1.1 Core Concepts in Minuku

Context Source—A *Context Source* is defined as a source of contextual information, such as location, activity recognition, accelerometer sensor, light sensor, application usage, batter percentage, Wifi availability, ringer mode, and so on.

Record— A *Record* is a piece of contextual information generated in Minuku. A Record has a *data* field to store information from a contextual source, and a

timestamp field to store the time at which the Record is generated. Because different contextual sources may have different numbers and formats of value to store in a Record, the data field adopts a JSON format[21], a open standard format for flexibly storing a set of unstructured attribute–value pairs, to store contextual information. A Record also stores a list of identifiers of *Session* that indicates which Session(s) this Record is associated with.

Session—A *Session* (or *Logging Session)* is a period of time during which Records are generated and logged. It is also a reference with which Records logged during the period are associated with. This reference is created because in Minuku, multiple sessions are allowed to run simultaneously, making it necessarily for Minuku to remember with which Session(s) a Record should be with associated. A Session also stores a Task Id, indicating which Task this Session is recording for.

Task—A *Task* (or *Study Task*) is a mission that researchers request users to accomplish in a study. For example, a Task being "record your location for 10 minutes at 9 PM for 12 days" is aimed to collect twelve 10-minute long logging Sessions, where each Session is associated with location Records from 9:00 PM through 9:10 PM.

State—A *State (*or *State* of a Context Source*)* refers to a condition of a Context Source, which is named and defined by the researcher in configuration with at least one criterion. For example, researcher can define a State named "At Home" for the Context Source Location with a criterion "the current location is

[21] https://en.wikipedia.org/wiki/JSON

within 50 meters away from the home location," where the home location is presented by a pair of latitude and longitude.

Situation—A *Situation* refers to a set of conditions Minuku monitors over time that are satisfied. For example, the State "At Home" can be combined with a State "Using Facebook" to create a Situation "Using Facebook at Home." The Situation is said to be detected when both States are satisfied.

Action—An *Action* is an act that Minuku executes to achieve a particular purpose, such as logging information of a Context Source, monitoring a Situation, creating a questionnaire, etc. An Action has properties including type, launch style, continuity, frequency, etc. An Action can be continuous or non-continuous (i.e. one time). Continuous Actions include logging and monitoring a Situation. If an Action is continuous, it has a state of *active* or *inactive*. When an Action is continuous and active, it is executed according to a designated rate. A continuous and *paused* Action is inactive. It returns to the active state after it is *resumed*.

Action Control (AC)—An *Action Control* is a control that Minuku uses to change the state of an Action. In Minuku, four Action Controls are currently available: *Start, Stop, Pause*, and *Resume*. A *Start AC* starts an Action. A started and non-continuous Action is executed immediately; a started and continuous Action is made *active* and put in a `RunningActionList`. Any action in this list will be executed based on a rate as long it is *active*, as mentioned earlier. A *Stop AC* cancels all scheduled instances of an Action (an Action can be scheduled to be executed multiple times). It also removes the Action from the `RunningActionList`, if it is a continuous Action. Finally, a *Pause* AC makes a continuous Action inactive and a *Resume* AC makes a continuous Action active. Finally, an Action Control is launched by Minuku according to some

rules, such as that it is launched when the Miniku application starts, is launched by a trigger, or is launched by a time-based schedule. In other words, In Minuku, it is the Action Control instead of an Action that is more often triggered. As a result, as a Start AC can be triggered by an occurrence of a Situation being detected, a Stop AC, a Pause AC, as well as a Resume AC can also be triggered by a Situation being detected. Moreover, AC is associated with a *Schedule*, which determines when and how often an Action should be started, stopped, paused, or resumed. As a result, the design of Action Control gives researchers more flexibility regarding when and which Action should take place.

Schedule—A *Schedule* defines when and how often Minuku should execute certain Action. A basic Schedule defines a sampling method and a delay. For example, researchers can specify that a Start AC starts an Action thirty seconds after the AC is launched and since then the Action is executed five times every hour. I will present more details later.

Context State Manager—A Context State Manager is a unit that extracts and logs, and manages the States of a class of Context Sources. Minuku currently have six Context State Managers implemented: `LocationManager`, `PhoneSensorManager`, `ActivityRecognitionManager`, `TransportationManager`, and `PhoneStatusManager`, and `UserInteractionManager`.

6.2 Main Features of Minuku

6.2.1 Enabling Concurrent Logging Sessions

The first important feature of Minuku is allowing multiple Logging Sessions running concurrently. This feature serves for two important purposes. First, it

allows researchers to have as many customized Logging Sessions as they want in configuration. For example, a researcher may be interested in users' locations at all times during the study, but is particularly interested in also capturing the application usage when the users are at home. In this case, the researchers can configure a continuous location logging at all time (says, Logging Task A) and another logging of application usage (says, Logging Task B) when the users are at home. When the users are at home, both Logging Sessions are running and stored separately. In addition, if researchers include location for both Logging Task A and B in configuration, the Location Records logged by Minuku when the users are at home are associated with both Logging Sessions. Second, enabling concurrent Logging Sessions also means that users can participate in several data collection tasks simultaneously. As aforementioned, same Records will be associated with any running Logging Sessions needing the Records. This allows Minuku to create just one copy of Records for multiple Sessions instead of making multiple copies, thus reducing the space needed to storing the data when the users participate in multiple data collection studies. As a result, allowing concurrent logging not only is essential for supporting researchers to design different logging tasks for their studies, but also reduces data storage on users' smartphones. Xiao et al. (Xiao, Simoens, Pillai, Ha, & Satyanarayanan, 2013) analyzes the large barriers of large scale crowdsensing studies and suggests that a major burden on mobile users participating in crowdsensing tasks is the need to install different crowdsensing applications on one mobile phone, which do not share data with one another but sometimes lock certain sensor. Thus, Minuku's allowing concurrent recordings potentially address this issue by enabling users to participate in multiple studies with only Minuku installed.

6.2.2 Supporting Monitoring and Detecting Customized *States* and *Situations*

Prompting context-triggered questionnaire to obtain users' in situ responses is increasingly common in behavioral research (Intille et al., 2003) Essentially, context triggering means a research tool prompts the users with a questionnaire when a target context users are in is detected. This allows researchers to collect users' experiences in and related to the detected context, such as after using the phone (Ferreira et al., 2014) or taking a bus (Froehlich et al., 2007b). In addition, researchers sometimes also log events that they want to inquire about later in a diary (Y.-J. Chang & Tang, 2015). Because different research would entail monitoring and detecting different contexts and events, some of which are complex and highly situated, it is important that a research tool claimed to support behavioral research support customizing "context" researchers desire to monitor and detect.

Minuku supports customizing "context" by allowing researchers in defining States and Situations respectively. As introduced earlier, State is a condition of a Context Source. A State can be simple as "being at home" (location), "using Facebook" (application), "taking a bus" (transportation), or the "phone is being charged" (battery); however, it can also be as complex as "neither at home nor in the office" (location) and "engaged in a video chat on the phone over 50% of the time in the last 30 minutes." (application). Minuku supports complex States as such by allowing researchers to specify a set of criteria for each State. Then, researchers can customize Situations by including one State or combining multiple States in the monitoring list. For example, researchers can monitor and detect a complex Situation such as "engaged in a video chat on the phone over 50% of the time in the last 30 minutes when the phone is being charged after 7 PM at home." (lets say, a Home-Phone-Video-Chat Situation). This Situation is detected when all the States included in the Situation are satisfied. Researchers

can define as many Situations as they want, and use these Situations to trigger actions relevant to this context.

6.2.3 Enabling Sophisticatedly Situated and Scheduled Actions

Minuku integrates customizable Situation, Action Controls, Schedules, and a Trigger framework that realize the execution of sophisticatedly situated Actions, which were not possible in previous research tools. For example, instead of monitoring the Home-Phone-Video-Chat all day, researchers can monitor "using video chat" all day first, and, when that Situation is detected, triggers monitoring the Home-Phone-Video-Chat Situation. Situating the latter monitoring Action can potentially reduce the battery consumption by only obtaining GPS location data when the users have been found engaged in a video chat. Similarly, the Home-Phone-Video-Chat Situation, when detected, can be used to trigger more Actions such as logging the status of the phone during the video chat, or prompting users with a questionnaire asking their experience related to the video chat. Situating Actions like such, therefore, allows collecting only the data relevant to the video chat behavior in the home environment. Furthermore, what makes Minuku novel from previous research tools is its framework called *m-Trigger framework* that permits triggers not only between Situations and Action Controls, but also among Action Controls themselves and others (more details later in section), such as triggering the questionnaire after the end of the logging session. This flexibility is designed with the goal for satisfying researchers' any needs in collecting specific data in specific situations. Finally, Schedules of Action Controls allow researchers to specify and constrain when and how often the execution of the Actions should take place. For example, researchers can delay data logging after the "Home-Phone-Video-Chat" Situation for 30 seconds, and schedule two questionnaires at random times in the next three hours after the video chat.

6.2.4 Configurability, Flexibility, and Extensibility

As previous tools claimed to be serve for a generic purpose, Minuku is also designed for high configurability, extensibility, and flexibility. Researchers can configure and customize States, Situations, Action Controls, and Schedules. They can also configure settings of Context Sources, phone notifications, questionnaires, and the backend server to submit data. Currently configuration is manually edited in a JSON file, which hides the complexity of Android system from the researchers. Thus, researchers do not learn and deal with Android programming to configure Minuku for their own study. In addition, as we discussed in the previous features, Minuku is highly flexible regarding the contexts to monitor and detect, and regarding the control of how to trigger and when to execute actions. Finally, Minuku is extensible by allowing adding new Context State Managers. Specifically, Minuku provides a model of Context State Manager that researchers can extend to implement their own Context State Managers. The model is flexible enough that it permits adding self-defined Context Sources and States. Thus, researchers not only can receive information from external sources such as an Android smart watch, but also are able to monitor Situations including the information from that source. This extensibility helps Minuku connect to emerging technology for obtaining a wider range of contextual information. Moreover, researchers can also create a new Context State Manger that further processes the information obtained from existing Context State Managers. `TransportationManager`, for example, is an instance of an extended Context State Manager created to further process the activity information generated from `ActivityRecognitionManager` to specifically monitor transportation related Situations. However, researchers need to equip knowledge of Android programming in order to create a new Context State Manager.

6.2.5 Supporting Participatory, Context-Triggered, and Hybrid Data Collection

Finally, as we proposed in the previous Chapter, Minuku supports performing Participatory, Context-Triggered, as well as a Hybrid approach—the combination of the Participatory and the Context-Triggered approaches. Minuku lets researchers configure whether they want to activate the Participatory approach or the Hybrid approach. If researchers choose to activate either one, a manual recording feature will become available in the Minuku interface. In the configuration for the Hybrid approach, researchers simply specify what Situations they want to detect in the Hybrid Approach. At runtime, Minuku will keep monitoring the specified Situations, and will prompts users to record and annotate the Situations as in the Context-Triggered In Situ approach. However, users' self-initiated recording and annotating action will trigger cancelling the prompted reminders of recording and annotation, respectively. With the simplified configuration, researchers can more easily employ each of these methods without manually define the relationships between Situations and Actions.

6.3 Implementation

Minuku is implemented in approximately 30,000 lines of code using the Android SDK[22] and the Google Play Service APIs[23]. It can run on Android devices between Android 4.1 and Android 6. Minuku uses a JSON format for configuration. It has five Context State Mangers that collects a variety of contextual information (show in Table 6.1), which can be stored locally on the device into a built-in SQLiet database or into the Android file system as text files. When a remote server is configured, Minuku can synchronize collected data with

---

[22] http://developer.android.com/sdk/index.html

[23] https://developers.google.com/android/

the server using either a GET or a POST request to transmit JSON documents. Synchronization can be configured to take place with a fixed frequency (e.g. an hour) or to take place only when the phone is connected to a Wifi network. Researchers can implement a web interface to process the requests to store data into a relational or non-relational database. Minuku is energy efficient in that it can stop the sampling or lower the sampling rate of Context Sources such as location when the phone is still and not being used. The lower sampled rate can be configured in configuration.

Minuku is designed to be used as a stand-alone application. However, it can also be used as a library within another application. I introduce the key components of Minuku in the following sections.

6.3.1 Extracting, Monitoring, and Logging Contextual Information

6.3.1.1 Context Manager

Context Manager is a core and the most central component of Minuku. It is mainly responsible for a) configuring and assigning tasks to Context State Managers, b) gathering and storing Records from Context State Managers, and c) monitoring specified Situations.

6.3.1.1.1 Configuring and Assigning Tasks

When the Minuku service first starts, Context Manager initiates all Context State Managers registered in Minuku, and inquire a list of Context Sources available in each Context State Manager. Context Manager itself does not keep a list of Context Sources so that it simplifies the addition and modification of Context State Mangers. Researchers only need to modify their own Context State Mangers without needing to revise any code in Context Manager. To add a new Context Manager, researchers only need to write one line of code:

`mContextStateMangers.add(mAndroidWatchManager).` Context Managers (re)assign two types of tasks whenever configuration is updated: *Logging Task* and *Monitoring Task*. Logging Task refers to logging data of a Context Source, and Monitoring Task refers to monitoring Situations. As researchers specify these tasks in configuration, Context Manager receives the tasks parsed by a Configuration Manger and then assigns them to the corresponding Context State Mangers according to the type of Context Sources associated with each task.

6.3.1.1.2 Gathering and Storing Records

Context Manager manages a unit called *Public Record Pool* to gather Records generated by Context State Managers. Records stored in the pool then will be saved to the local database and/or to the file system as log files, depending on the researchers' need. When Records are saved to the local database, Records are associated with the currently running Session(s) that request the Records. The Public Record Pool will note which Record has been associated with a Recording Session and saved into the database. It periodically removes saved Records to avoid accumulating too many Records.

Minuku manages two types of Logging tasks, Action Logging and Background Logging. Action Logging is an Action triggered or scheduled, based on researchers' configuration. Each this Action being executed creates a new Session with a unique identifier, making it easier for researchers to keep track of the occurrence of each logging. Background Logging is managed by Context Manager. It can neither be triggered nor scheduled; instead, it runs at all times when researchers activate it in configuration. The separation between Background Logging and Action Logging is to allow researchers to have an opportunity to capture both users' complete behavior history and users' behaviors only in

targeted situations so that researchers can treat these data independently in data analysis.

6.3.1.1.3 Monitoring Specified Situations

Context Managers keeps a list of Situations to monitor. Instead of using a "pull" approach—itself proactively monitoring Situations, Context Manager uses a "push" approach—it is notified by Context State Managers to monitor Situations. Specifically, Context State Manager notifies Context Manager whenever a State of its Context Source is changed. After being notified a State changed event, Context Manager finds all Situations involving this State and then examines States associated with each of those Situations. A Situation is said to be detected when all States involved in that Situation are met. Context Manager then consults Trigger Manager regarding what is trigger by the detected Situations.

6.3.1.2 Context State Manager

As briefly introduced earlier, a Context State Manager is a unit that manages Context Sources of a particular class. To speak more specifically, a Context State Manager defines a list of Context Source it manages, and has three major tasks to perform periodically. First, it extracts information of Context Sources that are *requested*. A Context Source is requested if and only if it is included in a logging task or a monitoring task in configuration. If a Context Source is not requested, it is deactivated and the Context State Manager does not extract the information.

The second task of a Context State Manager is storing data of the requested Context Source as Records in a *Local Record Pool*. Each Context State Manager has its own Local Record Pool. This Local Record Pool serves as a memory cache that a) allows the Context State Manager to monitor the States of Context Sources, and b) allows the Context Manager to copy the Records to the Public

Record Pool. Each Context State Manager has its own parameters to control the size of the Local Record Pool.

The third task of a Context State Manager is managing the States of Context Sources. In configuration researchers define a States of a Context Source by specifying a set of *Criteria*. A *Criterion* contains, at least, three properties: a *measure*, a *relationship*, and a *target value*. A Criterion is met when the current value of the measure matches the relationship between the its value and the target value. Below is an example of a State regarding whether the user is currently using Facebook on his or her phone. The measure **LatestUsedApp** prompts `PhoneStatusManager`, the Context State Manager that manages application usage, to find the latest Record of **PhoneStatus-AppUsage**. If the value of the data is "Facebook", which is **Equal** to for the target value **FaceBook**, `PhoneStatusManager` changes the State value to **Use Facebook**.

```
"State": "Use Facdbook",
"Source": "PhoneStatus-AppUsage",
"Value_Criteria": [
 {
  "Measure":"LatestUsedApp",
  "Relationship":"Equal",
  "TargetValue": "Facebook"
 }]
```

The State value is free text. Common measures and relationships are already predefined in the Context State Manager model. Thus researchers can use these measures and relationships directly in any extended Context State Manager (including their own ones). Predefined relationships include *"=", ">", ">=", "<>", "<", "<="* for numeric values, and "Equal", "Not equal", *"*Between", and "Contain" for textual values. Predefined measures include "LatestValue", "MostFrequentValue", and "MeanValue" (for numeric values). Researchers can add new measures and relationships to a Context State Manager. They can also

add additional parameters necessary for calculating certain measures. The example below shows a State of **At Home** for **Location**. Because it is necessarily to specify a target location (e.g. home) in order to calculate a distance, the Criterion below include a tuple of latitude and longitude as an additional parameter. The measure CurDistFromLoc is also a unique measured added to `LocationManager`.

```
"Id":3,
"State": "At Home",
"Source": "Location",
"Value_Criteria": [
 {
  "Measure":"CurDistFromLoc",
  "Params":
  [
     "42.293820,-83.701918"
  ],
  "Relationship": "<=",
  "TargetValue": 100
 }]
```

Finally, researchers can also include a set of Time Criteria when defining a State. For example, the following criterion specifies the value criteria has held true for at least ten seconds in order to make change the State.

```
 "Time_Criteria": [
{
 "Measure": "duration",
 "Relationship": ">",
 "TargetValue": 10
}]
```

When the value of a State is changed, the Context State Manager notifies the Context Manager to check if the Situations involving the State are met or not.

6.3.1.3 The Situation Monitoring Process

The entire Situation Monitoring process is illustrated in Figure 6.1. Generally speaking, the monitoring process consists of four steps: First, the Context State Manager determines the value of a State based on the specified criteria. Second,

the Context State Manager notifies Context Manager about the update, if the State value is changed. Third, after being notified, Context Manager checks all Situations involving the States, and for each Situation, check all States involved. Finally, Context Manager examines whether a Situation is met based on the States involved. After these four steps and if a Situation is detected, Context Manager calls Trigger Manager to find if anything (e.g. Action Controls) would be triggered.

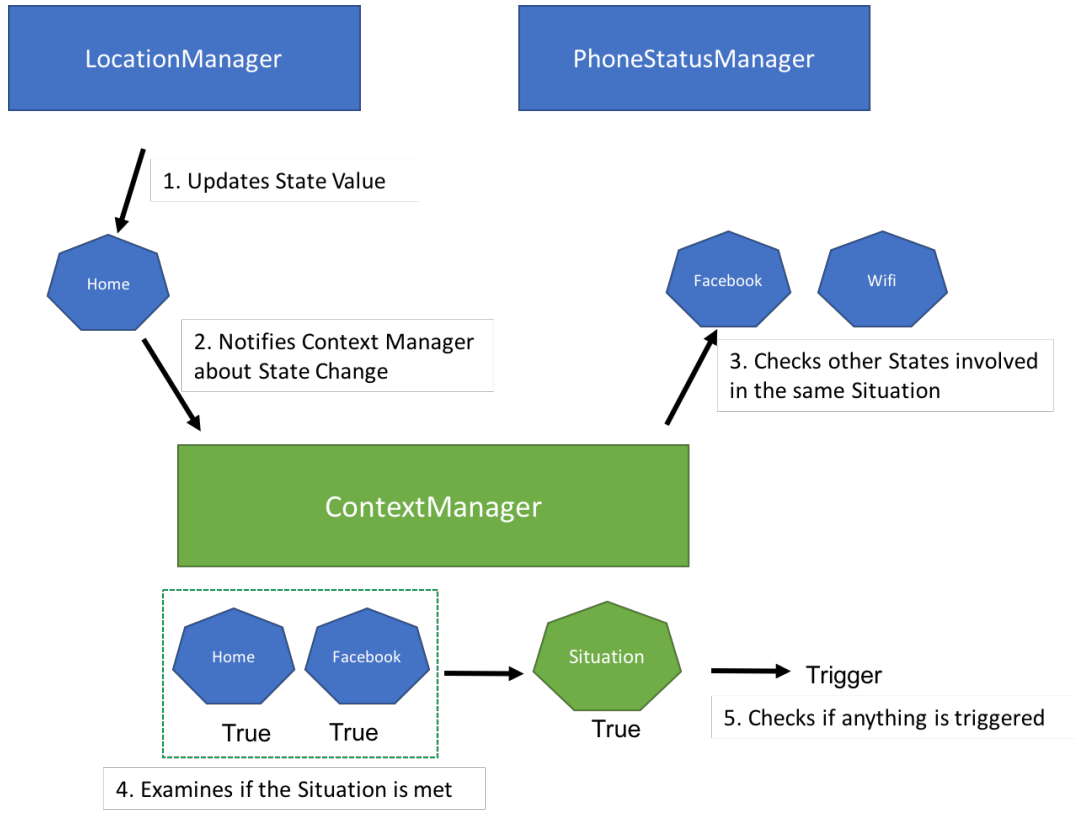


Figure 6.1 An example of the monitoring a Situation of using Facebook at Home, which involves two States: Using Facebook, and Being at Home.

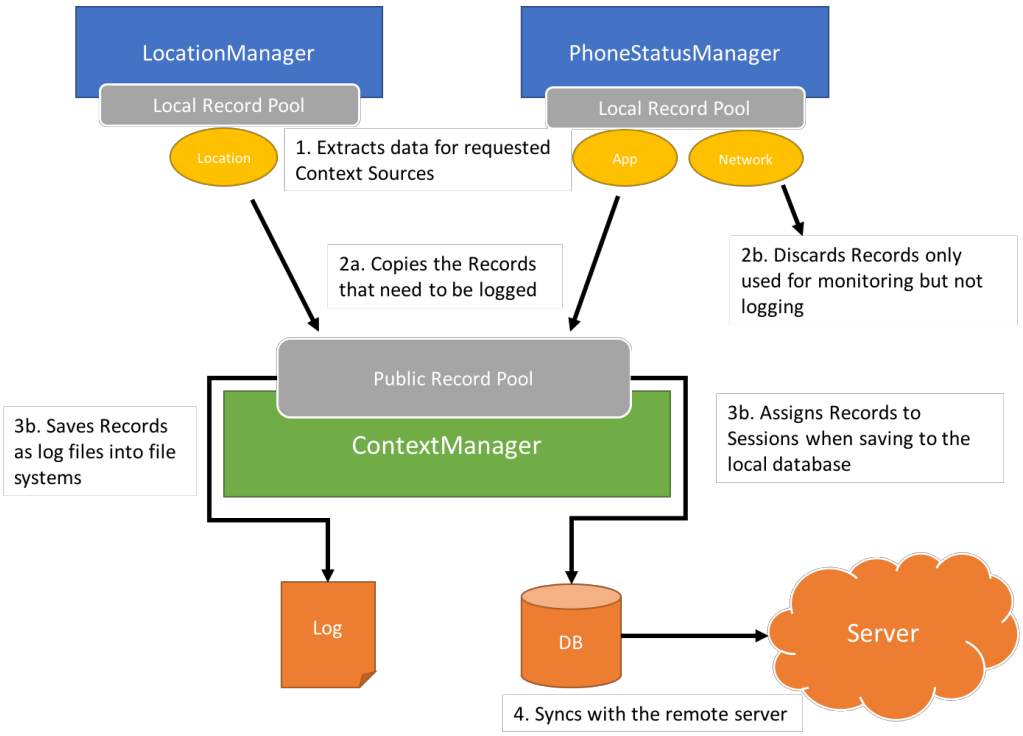


Figure 6.2 The process from extracting data, storing data to a Local Record Pool, copying data to a Public Record Pool, and saving data as log files or into a database.

6.3.1.4 Data Logging Process

The data logging process is illustrated in Figure 6.2. First, a Context State Manager extracts data of Context Sources depending on whether or not they are requested, either for logging or for monitoring Situations. Then, these data are stored as Records into the Local Record Pool of the Context State Manager. Second, the Context State Manager copies the Records in the pool to the Public Record Pool of Context Manager that need to be logged. Records used for monitoring are discarded when the Context State Manager refreshes the Local Record Pool. Third, Records in the Public Record Pool are stored into the file system of the phone as log files and/or into the local database, depending on researchers' configuration. Records stored into the databases are associated with

currently running Sessions that need the Records. Fourth, the local database synchronizes with a specified remote server using a GET or POST request.

6.3.1.5 Context Sources, Context Information, and Modes of Acquisition

Minuku supports acquiring more than 50 types of contextual information from 23 ContextSources. Table 6.1 below lists these types of contextual information (column 3) with the Context State Managers (column 1) and the Context Sources (column 2) they are associated with. Minuku uses different ways to acquire the information of these Context Sources, depending on their types. Specifically, Minuku uses a "push" method to obtain Context Sources including Location, Activity Recognition, Phone Sensors, and User Interactions. Location and Activity Recognition information is obtained through the Google Play Service, which automatically pushes information to the phone. Phone Sensors information is acquired using the sensor listeners provided by the Android system. The listeners use a callback function called `OnSensorChanged` to pushes sensor information. User Interaction information is in three categories. The first category is user interaction relevant to Minuku, including actions on the notifications generated by Minuku, actions related to recording and annotation, and actions in a questionnaire interface. The occurrence of these actions is directly obtained via a logger function in Minuku. These actions allow researchers to keep track of users' compliance in the study. The second category is user actions on the device, including tapping (clicking), pressing, swiping, and typing in an application, or pressing a physical button on the device. Acquiring information of these actions requires the AccessbilityService provided by the Android system. Because this service is highly intrusive and can monitor users' any actions on the phone, it requires the users to activate the service manually in the phone setting. The third type is the status of phone notifications. The information of notification is obtained by the AccessibilityService on Android phones before the version 4.3.

Context State Managers	Context Source	Content
LocationManager	Location	{ **latitude, longitude, accuracy, altitude, provider, speed, bearing** }
ActivityRecognitionManager	ActivityRecognition	{ [**activity**: **confidence]** } (labels: in a vehicle, on bicycle, on foot, walking, running, tilting, still, unknown)
TransportationManager	Transportation	**{transportation mode}** (modes: in a vehicle, on bicycle, on foot, static)
PhoneSensorManager	Accelerometer	**{x, y, z}**
	RotationVector	**{xsin, ysin, zsin, cos}**
	Gravity	**{x, y, z}**
	Gyroscope	**{x, y, z}**
	Light	**{light}**
	MagneticField	**{x, y, z}**
	Pressure	**{pressure}**
	Proximity	**{proximity}**
	AmbientTemperature	**{temperature}**
	RelativeHumidity	**{humidity}**
PhoneStatusManager	AppUsage	{**screen status, [application, last used time, duration]** }
	Ringer	**{ringer mode, audio mode, volume notification, volume ring, volume voice call, volume system, volume music}**
	Battery	{**battery level, battery percentage, charging source, is charging**}
	Telephony	{**operator name, signal type, signal strength, call state**}
	Connectivity	{**network type, network availability, network connected, wifi availability, wifi connected, mobile availability, mobile connected**}
UserInteractionManager (* indicates the requirement for the AccessibilityService; ** indicates the requirement of using AccessibilityService for Android phones below 4.3)	InMinukuAction	**{action, target, time}**
	Notification**	{**receive noti, select noti, dismiss noti**}
	InAppAction*	{**action, target, time, application**}
	OnDeviceAction*	{**action, target, time**}

Table 6.1 shows the current Context Sources supported by Minuku.

Finally, Minuku uses a "pull" method to acquire information of PhoneStatusManager and TransportationManager using a fixed rate (by default one reading per five seconds). For the latter, Minuku uses a finite state machine to process the information obtained from ActivityRecognitionManager to generate a transportation mode label periodically. More information about finite state machine is presented in the last section of Implementation.

6.3.2 Executing, Triggering, and Scheduling Actions

6.3.2.1 Action Manager

Action Manager is another core component of Minuku that manages Actions and Action Controls to control the behavior of Minuku. Although it is possible to make a variety of Actions available, the author chooses to focus on the Actions core to data collection, including:

- `MonitoringSituation`— Monitoring Situations the phone is in.
- `SavingRecord`— Associating Records with Sessions and saving them into the file system or in the local database.
- `Annotating`— Adding annotations to a specified Session.
- `GeneratingQuestionnaire`— Generating a customized questionnaire on the phone, with an option of prompting users with a notification.
- `GeneratingEmailQuestionnaire` — Generate a customized questionnaire via an e-mail form, with an option of sending the e-mail from a server or invoking an e-mail composing window.

These Actions make Minuku sufficient for performing most essential data collection tasks. However, as Minuku is aimed to be extensible, researchers can add Actions upon their need.

In configuring the Actions to perform by Minuku in a study, researchers specify what actions they want to execute, the *continuity* of the action, and whether to repeat and when to execute the actions. *Continuity* specifies whether an Action is continuous or not. A continuous Action is executed with a fixed frequency, which can be configured but by default is one execution per five seconds. Although `SavingRecord` and `MonitoringSituation` are probably the two most common Actions to run continuously, Minuku permits any Action to be continuous, keeping the flexibility for researchers to add any new Actions they want to run continuously. Similarly, `SavingRecord` can also be a one-time and non-continuous, such as taking a snapshot of the contextual condition at a specific moment, the method used in the study described in Chapter 4.

Regardless of the continuity of an Action, researchers can specify whether an Action is *one-time* or *repeated*. While one-time Actions are executed immediately, repeated Actions are scheduled. In Minuku, the difference between a *continuous* Action and a *repeated* Action is that the latter needs requires a *schedule*, for which a unit called Schedule Manager computes sampled times for the Action. A continuous Action, on the other hand, is executed upon a central clock of the Minuku service, taking significantly less computing and the system resource. Furthermore, the separation of these two allows researchers to run continuous Actions at specific times in a day instead of all day. For example, researchers may schedule continuous `SavingRecord` Actions at three random times between 10 AM and 10 PM, each of it lasts one hour. The configuration for this Action simply looks like below. Consequently, researchers have an option to decide between running a continuous Action or a repeated Action based on how frequently they the Action to be performed, or even, have an option to combine the two.


```
"Id": 1,
“Name”: “Randomized logging location and app usage in a day”
```

```
"Continuity":
{
   "Rate": 3,
   "Duration": 3600
},
"Execution_style":"repeated",
"Type": "monitoring_events",
"Logging_tasks":"1,2",
"Control":{
 "Start":
 [
  {
    "Launch": "schedule",
    "Schedule":
    {
     "Sample_method":"random",
     "Sample_count": 3,
     "Sample_startAt": "10:00",
     "Sample_endAt": "22:00"
    }
  }
 ]
}
```

To schedule and/or to trigger the execution of an Action, Action Controls (AC) are essential to specify in configuration, as shown in the example above (i.e. the Control property). As I introduced at the beginning of the Chapter, Minuku supports four types of Action Controls: Start AC, Stop AC, Pause AC, and Resume AC. The separation of Action Controls from Actions makes controlling the behaviors of Minuku flexible. It makes it possible for researchers to start, pause, resume, and/or stop the execution of any Action by schedules or by triggers. Researchers can also specify multiple methods to start, pause, resume, and stop the same Action. For example, the example below includes two ways to start a `GeneratingQuestionnaire` Action: 1) randomizing five times between 10 AM and 10 PM and 2) triggering by a Situation, with a sampling rate of 50% and with a five-second delay for the execution after the detection.

```
"Control":{
 "Start":[{
    "Launch": "schedule",
    "Schedule":
    {
     "Sample_method":"random",
     "Sample_count": 5,
```

```
    "Sample_startAt": "10:00",
    "Sample_endAt": "22:00"
   }
  },{
   "Launch": "triggered",
   "Trigger":
   {
    "Class":"Situation",
    "Id":2,
    "Sampling_rate": 50%
   },
   "Schedule":
   {
    "Sample_method":"simple_one_time",
    "Sample_delay": 5
}}]}
```

6.3.2.2 The m-Trigger Framework

One novelty of Minuku is its framework for supporting *Action-Trigger*, called *m-Trigger*. The mechanism of Action-Trigger is not new in data collection tools. It has been adopted in several previous tools that can deliver context-triggered questionnaires, as we have shown in Chapter 2. The core concept of Action-Trigger is that the users of the tool can define *Triggers* that fire the execution of specific actions. In previous tools, Triggers are associated with contextual conditions, which include sensor states, network status, user interactions, and so on. This framework allows users to specify what actions they want the tool to execute if a contextual condition is detected, which is generally useful for conducting an event-contingent ESM studies. The Action-Trigger mechanisms that previous tools adopted, however, are subject to a major limitation. That is, they are limited to *one order* of relations between a trigger and an action, which is mainly because *only contextual conditions can trigger an action*. This limitation hampers researchers from specifying more sophisticated behaviors of the tool, such as starting or stopping an action based on the state of another action. One example is "prompting users a questionnaire when a data recording is paused."

The *m-Trigger* framework breaks the limitation of only contextual conditions being able to trigger and only actions being able to be triggered. In Minuku, we defined an object class called *m-Object*. It is a parent class of all other objects in Minuku, including the Situation Class, Action Class, Action Control Class, and a Questionnaire Class. An instance of the m-Object-Class, i.e. m-Instance, such as a Situation, an Action, or an Action control, can trigger or be triggered by any other m-Instance. As a result, as a Situation can trigger an Action Control, an Action Control can also trigger a Situation (i.e. artificially creating a Situation). Likewise, users submitting a Questionnaire can also trigger an Action for logging data. In an even more complex relationship, an Action Control triggered by a Situation can further trigger other Action Controls and Situation, which, in turn, can trigger even more Situations or Action Controls using the same rule—forming a "trigger chain." This design also allows an m-Object to be a trigger of multiple m-Objects. For example, in the Context-Triggered In Situ condition described in Chapter 5, Miunuku monitors different transportation modes respectively, and then detecting either transportation mode, says walking, triggers: 1) logging data, 2) prompting users to annotate the walking trip, and 3) pausing monitoring walking. Consequently, the m-Trigger framework provides researchers with great flexibility to design sophisticated behavior of the tool and to performing highly conditioned actions. For example, researchers may specify "a Situation of being close to home triggers starting a `SavingRecord` Action for 30 minutes and pausing the `MonitoringSituation` Action. Then the end `SavingRecord` Action triggers a `GeneratingQuestionnaire` Action and resuming the `MonitoringSituation` Action." In summary, the m-Trigger framework is novel in that it does not limit a Trigger to contextual conditions (i.e. the Event in Minuku), but instead, allows Situations and Action Controls to mutually trigger one another, enabling researchers to flexibly design more sophisticated and situated behaviors of the tool.

6.3.2.3 Trigger and Schedule Manager

Trigger Manager is the core component that implements the m-Trigger framework. It maintains a list of *TriggerLinks*, each of which stores the information of a Trigger, a target to be triggered (e.g. a Situation, an Action Control), and a *trigger rate*. By default, a trigger rate is 100%. But it can be configured to any rate. For example, a researcher may only want to sample 20% of the phone call events on the phone for sending a questionnaire to the user. In order to support participation in multiple studies, each TriggerLink is only accessible and visible to the configuration where it is defined, so that TriggerLinks in different studies do not interfere one another.

Once Trigger Manager identifies which target to trigger, it passes Schedules associated with all triggered targets to *Schedule Manager* for computing when and how often Minuku should perform the task on the target, for example, stopping a recording action five minutes later, or launching a questionnaire three times randomly in next five hours. A *Schedule*, introduced earlier, can be configured with the following properties: sampling method, sampling delay, sampling interval, sampling duration, number of samples, earliest sampling start time, latest sampling end time, and minimum sampling interval. The inclusion of these properties depends on the sampling method a Trigger uses. For example, a *Simple-One-Time* method only needs to include a sampling delay. Other sampling methods including *Random*, *Random-with-Minimum Interval*, and *Fixed-Interval* need to include other properties. Schedule Manager computes the schedules for performing tasks on a daily basis. By default, the latest time that can be scheduled is 11:59:59 PM. Minuku refreshes all schedules at 4 AM and re-computes schedules that need to be repeated for the coming day.

With the combination of Action Controls, the m-Triggered framework, and Schedules, Minuku provides great flexibility and configurability, and support

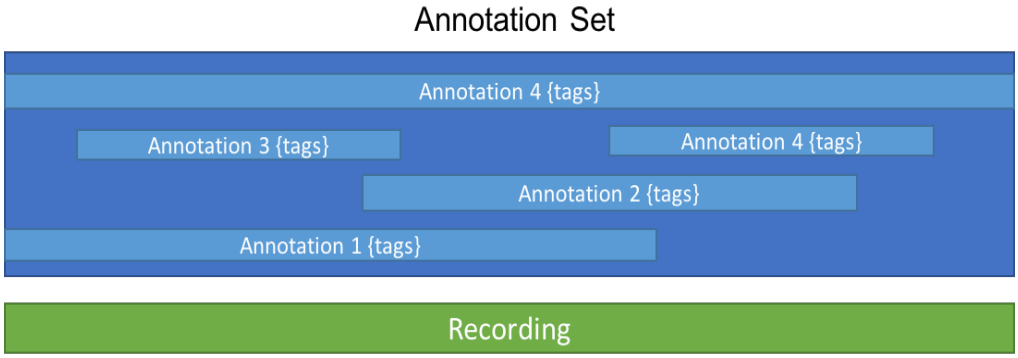


Figure 6.3 The Annotation Set Framework in Minuku

researchers in scheduling and situating data collection tasks with significantly less limitation compared to previous research tools.

6.3.3 Annotation and Recording

Annotation and Recording Manager is responsible for managing recordings and *Annotations* added to the recordings. Minuku has a flexible and extensible framework for adding Annotations. Specifically, an Annotation can apply to an entire recording or a specific sub-section specifying a start time and an end time (shown in Figure 6.3). Each Annotation has a textual content, and can be attached a list of tags as metadata describing the Annotation. For example, researchers can specify why the Annotation is added, who adds the Annotation, and what type of Annotation it is (e.g. a label, a comment). A recording keeps a reference of an *Annotation Set* that stores all Annotations for that recording. Thus, any Annotations added to the recording are added to the Annotation Set that is referenced by that recording. An Annotation Set is stored in a specialized data structure and is exported to JSON document when it is saved into the local database.

6.3.4 An Example of Extended Context State Manager: Transportation Manager

As mentioned earlier, TransportationManager is a ContextStateManager that processes activity labels from ActivityRecognitionManager and then generates new transportation mode labels using a finite state machine (FSM). The purpose of building this TransportationManager is to make Minuku able to capture, monitor, and detect users' mobility more reliably. As shown in Table X, ActivityRecognitionManager receives activity labels from the Google Play Service. Each label is accompanied by a confidence value, indicating how confident the service thinks the user, or the phone, is performing that activity. The service may include one or more activity labels, of which the total of all confidence values adds up to 100. For example, {*in_vehicle:100}* shows that the service is 100% confident that the user is in a vehicle; {*in_vehicle:77, on_bicycle:23}* shows that the user is most probably in a vehicle, with a small likelihood of biking. These results, generating through an activity recognition algorithm, however, are subject to positioning errors and noises. They are also affected by some unexpected or errant traffic conditions, making these labels often correctly reflect the user's actual transportation state. Thus, instead of directly using these labels, we build a TransportationManager that further process these labels using an FSM. At a high level, TransportationManager considers both current and previous activity labels obtained in a certain window size to determine the user's current transportation mode. The purpose of "looking back" at previous labels is to *raise the threshold of transitioning users from one transportation state to another state* so that the transition would be more resistant to noisy labels.

Figure 6.4 shows the process of the transportation FSM. Initially, a user is in a *Static* state. When TransportationManager receives an activity label indicating movement (e.g. on foot, in a vehicle, and on a bicycle), TransportationManager moves the user to the *Suspect Start* state, meaning that TransportationManager is

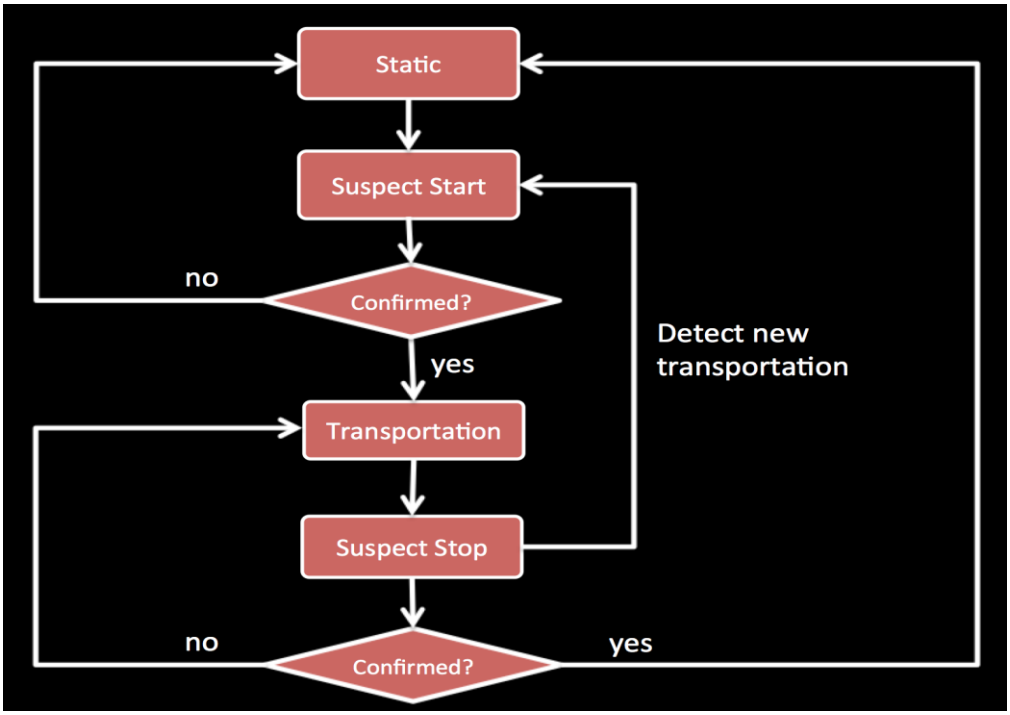


Figure 6.4 . The final state machine of the TransportationManager has four states: Static, Suspected Start, Transportation, and Suspected Stop.

suspecting that the user is moving in that transportation mode. When TransportationManager reaches this state, it waits for a period of time (e.g. 20 seconds), and then examines all previous labels within that time period. If higher than a percentage of previous labels agree the suspected transportation mode, TransportationManager moves user to the *Transportation* state, and informs Minuku that the user is in that transportation mode. When TransportationManager enters this states, it starts to look for labels indicating non-movement (e.g. *still*) or another transportation mode. It enters the *Suspect Stop* state if it receives any such a label. Similarly, TransportationManager returns to the Static state if over a period of time higher than a percentage of previous labels are either still or are indicating another transportation mode. However, if TransportationManager keeps receiving labels of another transportation mode and passes the threshold, it skips the Static state and directly enters the *Suspect Start* state of that new mode.

There are four important parameters to control the transition between the states: the durations of the periods in which TransportationManager checks activity labels for starting and stopping a transportation mode, respectively, and the thresholds for confirming a start and a stop of a transportation mode, respectively. All of these thresholds are arbitrarily set initially, but are tested and modified iteratively over a 4-week testing period. We stopped testing the accuracy of TMD while we reached the point at which it was sufficiently accurate for detecting a start and a stop of a TM.

Finally, we also built a small component called Mobility Manager, which simply distinguishes the user's mobility state between *static* and *moving* based on the transportation information from TransportationManager. MobilityManager considers a user *mobile* when TransportationManager generates a Record showing that user is in a certain transportation mode. Otherwise, MobilityManger considers the user *static*. The purpose of disguising between mobile and static is to allow Minuku stop the sampling or lower the sampling rate of power-expensive sensor such as GPS when the phone has been static. Previous works have shown that using an accelerometer to detect moving for knowing when is a good time for obtaining location data can significantly reduce power consumption. Essentially, the Google Play Service uses accelerometer to detect still, tilting, and walking. Thus, using the mobile/static information derived from activity labels can also make Minuku energy efficient.

6.3.5 Questionnaire Generation

Customizable questionnaires are necessary to allow researchers to ask different types of questions in Minuku. In Minuku, Questionnaire Manager is responsible for storing questionnaire templates and creating a questionnaire instance when Action Manager executes a `GeneratingQuestionnaire` Action. In configuring a questionnaire, for each question researchers want to include, they

specify an index, the type of question, and the option items for the question. Minuku currently supports open format question (textbox) and multiple-choice questions (checkbox and radio button). It automatically generates a textbox when researchers include an "Other" field. The questionnaire configuration creates a questionnaire template. When Action Manager executes a `GeneratingQuestionnaire` Action, it finds the specified questionnaire template and based on it creates a questionnaire instance—an instance containing a unique identifier, a set of questions, a set of responses associated with each of the questions, questionnaire generation time, attendance time, and submission time. Questionnaire, whenever is generated, is saved into the local database. The configuration below shows an example of questionnaire.

```
"Id": 2,
"Title": "Where do you place your phone?",
"Description":"Please answer the following questions.",
"Type":"activity",
"Questions":
[{
  "Index": 1,
  "Type":"textbox",
  "Question_text": "Where are you now?"
 },
 {
  "Index": 2,
  "Type":"multichoice_one_answer",
  "Has_other_field":true,
  "Question_text":"Where did you just place your phone?",
  "Option":[{
    "Option_text":"Desk/Table"
   },{
    "Option_text": "Pocket"
   },{
    "Option_text": "Bagpack/handbag"
   }]
 },
 {
  "Index": 3,
  "Type":"multichoice_multiple_answer",
  "Question_text": "What is/are the reason(s) that you place your phone there?",
  "Option":
  [{
    "Option_text":"It's easier to notice notifications."
   },
```

```
    {
     "Option_text": "It's convenient to grab."
    },
    {
     "Option_text": "It's less disturbing."
    }
   ]
  }
 ]
```

Figure 6.5 shows the resulting questionnaire. When the user opens the questionnaire, Minuku updates its attendance time. After the user submits the questionnaire, Minuku updates its submission time and the responses. Finally, Minuku is also capable of embedding data such as detected events into a questionnaire. However, current this feature is not yet available for configuration.

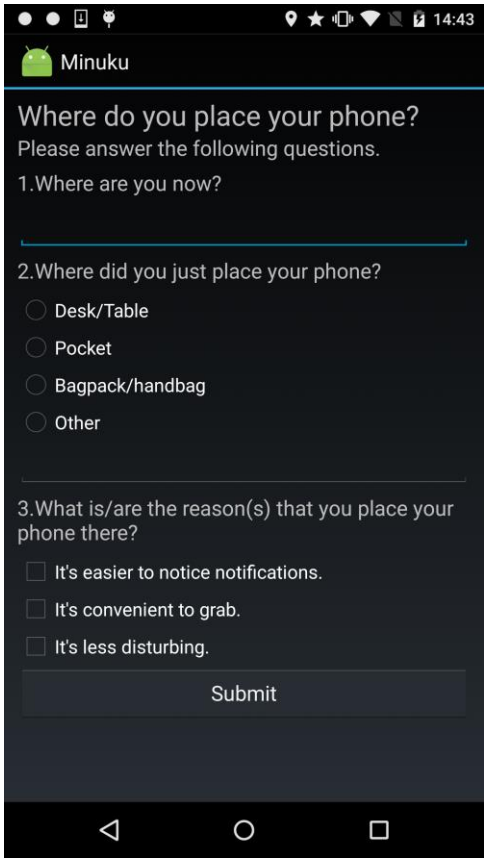


Figure 6.5 An example of customized questionnaire.

6.3.5.1 Status Tracking and Debugging

One common concern researchers have collecting data on smartphone users' phones is: *Is the system running and collecting data as expected*? Few previous research tools are equipped utilities for researchers to track the status of the system. As a result, researchers can only infer the status via tracking the data been uploaded to a server. However, checking data is not a reliable way to infer the status of the system in real time. This is particularly true when the research tool is configured to upload data only when the phone is connected to a Wifi network: when smartphones users are on the go, their phones are very likely to use a mobile cellular network instead of a Wifi.

Minuku allows researchers to track the status of the system in two ways. First, it periodically sends a POST request to the configured server that indicates the service is running, regardless which network the phone is connected to. The request only contains a small piece of information, including a user identifier, the

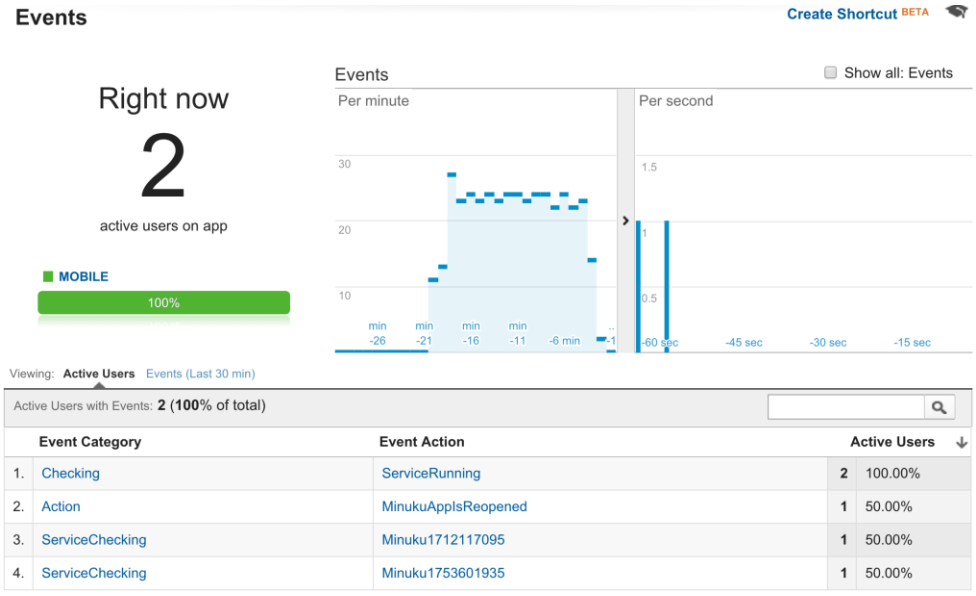


Figure 6.6 The Google Analytic allows researchers to track whether the Minuku service is running on participants' phones.

study the phone is in, and a timestamp. Second, Minuk uses a Google Analytic[24] service that allows researchers to log-in the Google Analytic dashboard using their own account (shown as Figure 6.6). Using the same frequency in the first method, Minuku fires an event to the Google Analytic service with a message containing the information of user identifier (the content of the message will become configurable in the future plan) Using either utility, or even both, researcher can track the status of every single phone participating in the study. Finally, Minuku also uses the a service called Fabric Crashlytics[25] that informs researchers about crashes of Minuku. A crash report is sent to a specified email address, which provides a link to the dashboard of the service.

With these utilities for tracking the status of Minuku in real-time, Minuku allows researchers to discover and diagnose issues promptly, and accelerate communication with a technical support of Minuku (if there is any in the research team), and with study participants for informing them with further instructions (e.g. restart the Minuku app).

6.3.6 Configuration of Minuku

As I have shown in previous sections, all of the major functionalities of Minuku are configurable by the researchers. When researchers update a configuration and restart the Minuku service, Configuration Manager parses the configuration and updates the existing configuration of Minuku. Each study (or project) can have its own configuration. Things that can be configured include Tasks, Context Source Settings (e.g. sampling rate, sampling mode), Backend Remote Server, Background Logging, Logging Tasks, Context Source States, Situation, Actions,

---

[24] https://www.google.com/analytics/

[25] https://docs.fabric.io/android/crashlytics/introduction.html

Action Controls, Notifications, Schedules, Questionnaires, and other more specific items. Currently, configuration is processed when the Minuku service starts. In the near future, configuration can be configured from a remote server. This will allow researchers to modify the behavior of Minuku remotely during the study if necessary. The architecture of a Minuku that can be updated from a remote server is shown in Figure 6.7.

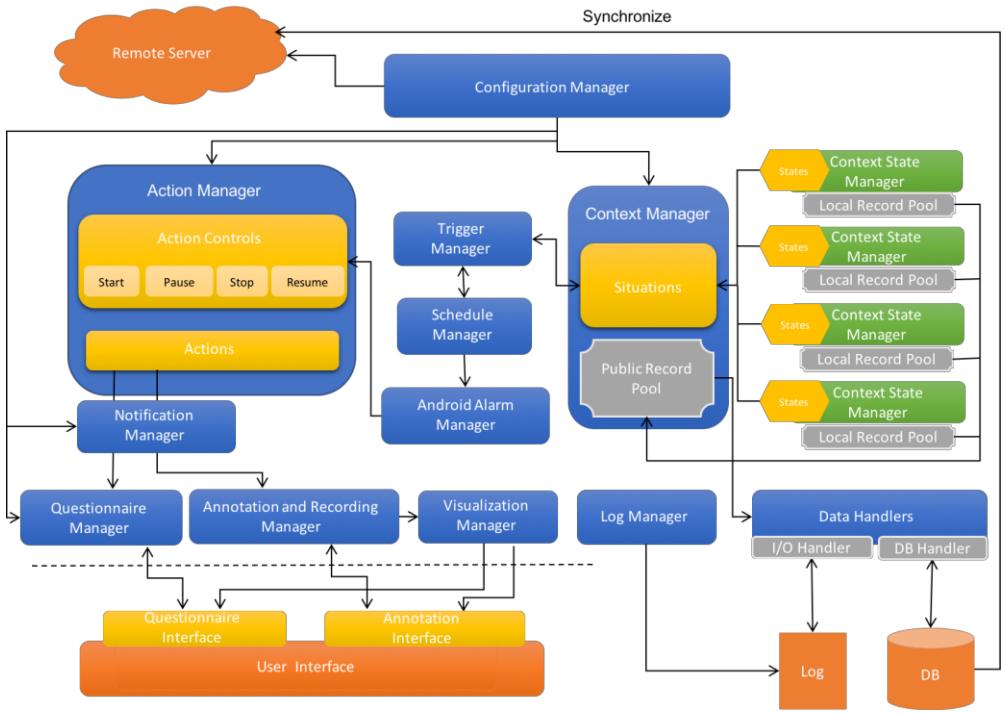


Figure 6.7 The architecture of Minuku. Yellow indicates an m-Object. Blue indicates a processing component. Green indicatres a Context State Manager. Grey indicates a unit processing data. Organge indicates an external unit.

6.4 Case Studies

Since Minuku was developed, it has been used in research studies for collecting data. Specifically, two research projects, including the study described in Chapter 5, used the first version of Minuku. Currently, Minuku is under testing for three ongoing research projects to be conducted in 2016 Spring. It will also support several projects in the near future.

6.4.1 Previous Projects

Minuku has been re-architected since 2015 Fall. I consider the version before this re-architecture Version 1. Minuku Version 1 was used in two research studies. The first study is presented in Chapter 5, in which Minuku was used to implement a participatory approach and two context-triggered approaches for collecting annotated travel data. For context-triggered approaches, Minuku's m-Triggerd framework was utilized to trigger the monitoring action of different transportation modes, to trigger logging, and to trigger in situ questionnaires. Minuku also scheduled fixed-time actions: prompting users to annotate in the POST condition at 9 PM, and a daily email questionnaire with data embedded scheduled at 9:30 PM. Regarding data logging, Minuku ran a Background Logging session to passively log location and activity during the study. Participants' recordings were generated by Action Logging, initiated either by Minuku via the context-triggered method or by the participants. In another study, however, Minuku was not fully utilized as it was for the aforementioned study. The goal of the study (C.-C. J. Huang, Yang, & Newman, 2015) was to develop models to predict households' thermal comfort. The researchers deployed sensing system at people's home and used Minuku to deliver ESM to obtain participants' thermal comfort sensation, comfort sensation, current activity, indoor location, clothing level, and brief notes that might help them recall the reasons for their sensation and comfort report when completing the end-of-day diary entry.

6.4.2 Ongoing and Future Projects

The current version of Minuki is under testing for three ongoing projects. The first project is a field experiment examining the effectiveness of a Hybrid data collection approach proposed in Chapter 5 for collecting annotated mobility data. In this project, Minuku will implement a Participatory approach, a Context-Triggered In Situ, and the Hybrid approach. The setting of the experiment is largely a replication of the study described in Chapter 6—comparing the effectiveness of the three approaches in the same study setting and conducting the same analysis. Additionally, Minuku will log contextual data used for correlating with participants' data collection behavior. The second project is an ESM study investigating how the use of location-based friend finder application enhances users' social capital. Minuku will implement randomized ESM questionnaires and passively log contextual information. The third ongoing project also will not fully use the capability of Minuku. The goal of the study was twofold. First, they aim to explore the relationship between assumed capacity to vary cadence—steps per minute, indicated by their prescribed K- level, and performance in daily life. Second, they aim to assess the degree to which subjects of certain K-levels exhibit signs of activity outside the home. Therefore, the researchers hope to use Minuku to passively monitor and log participants' mobility information, including location, activity labels, and transportation labels during the study period. All of these studies have their own specific needs for data logging and a designated server for uploading the data.

In the near future, Minuku is anticipated to be involved in several upcoming studies: a) a study exploring the role of different contextual features for modeling users' receptivity to interventions of performing a short-duration physical activity. Miunku will implement ESM questionnaires to collect ground truth of users' receptivity as well as passively log contextual data for behavior modeling. b) A study investigating contextual conditions that disrupt routine diabetes self-

management. Minuku is likely to implement a diary study with photos embedded and to perform passive logging and monitoring location. c) A study investigating users' psychological states after the use of social media on mobile phones. Minuku is expected to implement a context-triggered ESM that samples the use of mobile social media. It will also perform passive logging of contextual information. Thus far, which specific features of Minuku will be used in these projects have not been finalized. However, it is believed that through supporting these projects, it would become clearer regarding what features and functionality Minuku is still lacking and needs to add to make it a more full-fledged research tool. Minuku is likely to have more features, configuration items, flexibility, or extensibility after supporting these studies.

6.5 Discussion

6.5.1 Contributions

Although Minuku is still an ongoing project and will add more enhancements in the near future, its several features has surpassed previous research tools. First, Minuku allows researchers to monitor complex contextual conditions by separating States from Situations. Researchers can include as many criteria as they want to map measures of a Context Source to States; then they can combine various States into a Situation for monitoring. Second, the combination of Action Controls, the m-Trigger framework, and the schedule of Actions make it possible to perform highly situated actions at different times. In addition, the m-Trigger framework of Minuku let triggering be not limited to one-order and to only between detected context and starting an action. Third, the elements involved in the two aforementioned features all have flexible configurable. In addition, the extraction of Context State Managers from Context Manager allows researchers to add their own Context State Managers to obtain contextual data from external resources. This extensibility make Minuku's stay timely even after new technology emerges. Finally, Minuku provides other features additionally

supporting researchers in using Minuku for collecting data through the mobile crowd. The features include permitting concurrent logging that allows participation in multiple studies, and allowing choosing a participatory, a context-triggered, or a hybrid approach to collect activity data. After Minuku is used for supporting more research studies, Minuku is likely to add more configuration items and have more features to become a more generic research tool.

6.5.2 Limitations of Minuku

Despite the contributions mentioned above, Minuku currently is subject to several limitations that need to be addressed in the future. It is noteworthy that these limitations, are also lacking in most, if not all, previous research tools.

6.5.2.1 Limited visualization

First, Minuku currently only provides a map for visualizing location trace of participants. In order to support different types of crowdsourcing tasks where showing location traces is not helpful (e.g. collecting activity in the home environment), Minuku needs to support more types of data visualization to facilitate annotation on the data. In the case where showing location traces is helpful, visualizing the characteristics of the traces such as speed or a transportation mode may better assist users in recalling those activities when annotation.

6.5.2.2 Questionnaire Design

Minuku currently supports only a limited set of types of questions. Including more UI components to support more types of questions would be desirable to enrich the questionnaire as well as to collect more types of responses (e.g. using a slidebar to obtain a numeric measure as an alternative for using radio buttons to obtain an ordinal measure). Furthermore, Minuku does not support the configuration of skipping and branching questions. It also does not support adding

objects (e.g. visualization, pictures) to a questionnaire. These features would be important to include as they may be essential for some research studies.

6.5.2.3 User Interaction

Minuku has a simple but very limited interface. When it is used as a data collection tool in mobile crowdsourcing, it would be worthwhile to enable users to perform more actions to review, manage their own data. It may be also worthwhile to allow in-app messaging to facilitate communication between users and the researchers. Moreover, researchers may also have a need to customize their own interface to interact with the users.

6.5.2.4 Limited Intelligence

Minuku is also limited in terms of intelligence. Although Minuku supports context-triggering, the context detection of Minuku is not based on a machine learning model for active learning (Stikic, Van Laerhoven, & Schiele, 2008). As a result, Minuku is not able to function like tools as Fisher & Simmons (2011) to improve its accuracy of detection for contextual conditions that can only be inferred (e.g. the user is eating) instead of directly observed (e.g. the phone is being charged)

6.5.2.5 No Web Dashboard Available

Minuku does not have a web dashboard for researchers or users to track, monitor, and manage the data. Instead, Minuku currently relies on the researchers to perform these actions using the dashboard of the remote server they choose to upload data for the study. However, having a web dashboard not only may let researchers and users more efficiently manage their data, but also allow researchers to configure Minuku remotely using a web interface they are more familiar with and may be more comfortable with using. It can also allow researchers to send messages to users participating in a certain study.

6.5.3 Flexibility and Configurability

While it is important for research tool to be configurable and flexible, it is noteworthy that there exists a tension between configurability and flexibility. Making the tool highly flexible would require greater complexity in the system design as well as in the configuration. That is, greater flexibility may mean more configurations items researchers will be exposed to and to modify. As a result, making everything configurarable is not ideal due to the risk of making researchers overwhelmed by the number of items. On the other hand, making the tool less configurable is likely to fail to fulfill researchers' need for their research. One benefit of using JSON for configuration is that researchers can include only the relevant configuration items without needing to include all, and Minuku will process those included ones and uses a default for any other non-included items. For example, although researchers can change the sampling frequency of all Context Sources in configuration, they can also choose not including any of them in configuration. Nevertheless, when Minuku adds more configurable items after adding more features that are currently lacking in the future, it may become necessary to design a "configuration wizard" that guides researchers in walking through the configuration items if configuration is made online. A guide to efficiently navigating through the documentation is otherwise necessary for offline configuration (i.e. JSON).

6.5.4 Future Work

As mentioned earlier, Minuku is under development and will add more enhancements through supporting more research studies. However, it would be important for Minuku to include several features regardless of the requirements of those case studies. My first near-future plan includes building more Context State Managers as a demonstration for researchers' reference. These include wearable devices such as Android smartwatches, third party wearable sensor wristbands (e.g. Angel Sensor), wearable cameras, and external websites (weather site), as shown in Figure 6.8. The second plan is to develop a web dashboard for

researchers to configure Minuku remotely as a basic feature. The third plan is to improve the in-app interaction, including supporting more visualizations in both annotation interface and in a questionnaire, and supporting more UI components in a questionnaire. Finally, it is important to document all possible configurations and extension as well as design a guideline for configuring a study.

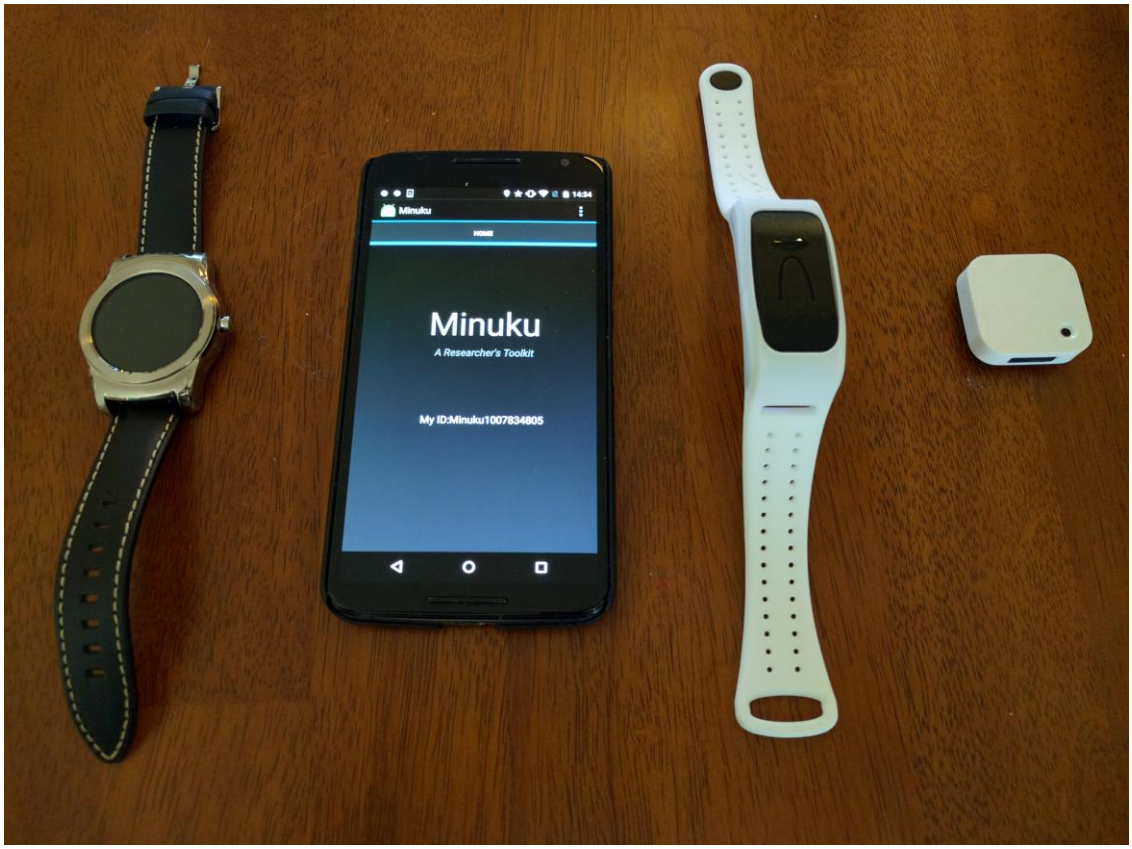


Figure 6.8 A future plan for Minuku is to develop Context State Managers for obtaining data from wearable devices such as Android watches (leftmost), wearable sensor wristbands (second rightmost), and wearable camera (rightmost).

Chapter 7 Conclusion

At a higher level, this thesis attempts to answer two particular research questions. First, what challenges designers and developers would face when using a capture-and-playback (C&P) approach and tool to design and develop context-aware applications, and what features are useful and essential to address these challenges to better support the design and development of context-aware applications? Second, what would be a good practice to collect annotated behavioral and contextual data via mobile crowdsourcing, and what features a capture tool should equip to make data collection more effective?

This thesis describes my research efforts in answering these questions. I address the first question via two case studies and a developer study of context-aware applications involving using a C&P tool called RePlay (Chapter 3). I reflect on the experience in the two case studies using RePlay and highlight three key activities a C&P tool should support in the developer study. Then I describe a new C&P tool called CaPla, built based on the findings from the developer study, and I investigated the effectiveness of the proposed features by evaluating CaPla.

I address the second question via two empirical studies. First, I conducted a field study to investigate the effectiveness of three different approaches for collecting annotated travel activity data via the mobile crowd. I conducted a pros and cons analysis and a user behavioral analysis. Based on the findings, I provided design and methodological suggestions for an ideal

approach and tool to effectively support collecting annotated behavioral and contextual data. The study also provided insights into users' receptivity to annotation tasks. The other study is an empirical research investigating mobile phone users' interruption management practices and how their practices affect their receptivity to incoming communication requests. The results of the research suggest how smartphone users' receptivity differs according to the ringer mode. In the rest of this chapter, I first highlight the key results and contributions from the three studies aforementioned. Then taking these together, I envisage a C&P infrastructure for achieving effective use of a C&P approach to develop context-aware applications. I also propose a number of new research questions for future research in context-aware application development to address.

7.1 Summary of the Results

Chapter 3 describes my research efforts specifically aiming to answer two research questions crucial to context-aware application development: a) what challenges designers and developers would encounter when using a C&P approach and a tool to design and develop context-aware applications; b) what kind of support a C&P tool should provide to address the challenges to make the C&P approach more effective. The goal is to inform the design space for C&P tools by answering these two questions. I followed two directions to answer these questions. First, I and my collaborators undertook two design projects of context-aware applications, in which we executed one full circuit of the interaction design lifecycle for each project and used RePlay to prototype, test, and evaluate the two systems. The goal was to reflect on our own experiences in exploring the benefits of, and the challenges in using a C&P approach and tool to design and develop location-aware systems. The results suggest that using a C&P approach and tool is

beneficial to prototyping, testing, and evaluation of context-aware applications at least in: helping answering design questions; examining design alternatives; testing features and algorithms; and creating realistic conditions for engaging participants in system evaluation. The aim for the second direction is to inform the design space of C&P tools through investigating developers' needs and behaviors in using a C&P tool (in this case, RePlay) to test and evaluate a context-aware application. Our results suggest three important activities a C&P tool should support: selecting examples, modifying data, and control playback during iterative testing. We then built a new C&P tool called CaPla, with features aimed for supporting these activities. Our evaluation of CaPla showed that CaPla effectively supported developers in making sense of captured data and in selecting good examples for testing a location-aware application. Throughout these two directions, we summarize three major challenges designers and developers would encounter in using a C&P approach and tool that need to be addressed in future work: a) the challenge of possessing the data needed for various development activities; b) the challenge of knowing what data is available for use for different development activities; and c) the challenge of selecting and creating suitable examples among a large amount of captured data.

Chapter 4 describes an empirical study investigating mobile phone users' ringer mode usage for managing interruption on the phone and how ringer mode usage affects receptivity—attentiveness and responsiveness—to incoming communication requests. I and my collaborator conducted a two-week empirical study with 28 Android smartphone using a mixed methods approach: using phone logging, diary study, interviews, and post-study survey to understand their real usage of the phone for communication, ringer mode changes, and qualitative experiences. Our results include two highlights. First, mobile phone users have diverse ringer mode usage, but they switch ringer mode for three main purposes: 1) avoiding interruption, 2) preventing their phone from disrupting the

environment, and 3) noticing important notifications. Second, ringer mode mainly influences attentiveness but not the responsiveness to attended messages. Third, without signals of notifications, mobile phone users are less likely to immediately attend to SMS messages than when with signals. In addition, mobile phone users are less attentive and responsive to SMS at certain locales. We provide design implications for future intelligent notification systems.

Chapter 5 describes a field study investigating using three approaches for collecting annotated travel activity data via the mobile crowd in real-world settings. The three approaches being compared are Participatory (PART), Context-Triggered In Situ (SITU), and Context-Triggered Post Hoc (POST). 37 Android users were recruited to use these approaches to collect their personal travel activity data when they were traveling outdoors using the first version of Minuku. We conducted two phases of analysis on the dataset. In Phase One, we compared the quantity and quality of collected data among the three approaches as well as participants' subjective experience in using each approach. Our results suggest two highlights. First, the data collected using the PART approach are more complete, contained less noise, and led to greater data coverage than those collected using the SITU and POST approaches. Second, while participants appreciated automated recording and reminders for their convenience, participants highly valued having the control over what and when to record and annotate. This suggests that user burden and user control are two important aspects a future tool in mobile crowdsensing/sourcing should take into consideration.

In Phase Two, we investigated how participants used Minuku to perform the PART and the SITU approach in the field to collect activity data, respectively. We particularly examined how the specific nature of the activities being collected affected their recording and annotation behaviors. In addition, we also analyzed the characteristics of participants' annotations to understand whether annotations

differed according to the type of activity being collected, and analyzed the diary entries to understand the reasons for unlabeled, mislabeled, and erroneous data. Our results showed that type of activity being captured influenced the timing of recordings and annotations, participants' receptivity, and characteristics of annotations. Moreover, these factors were impacted by the nature of transitions between activities, the attentional requirements of each activity, and the context of the activity.

Chapter 6 describes a mobile data collection called Minuku aimed to enable researchers to collect various types of behavioral and contextual data according to their needs. Minuku provides four important features that are beyond existing data collection tools. First, Minuku allows researchers to monitor complex contextual conditions. Second, Minuku introduces Action Controls, a m-Trigger framework, and sophisticated schedule of Actions, making it possible to perform highly situated actions at different times. Third, Minuku is configurable, flexible, and extensible. Finally, Minuku allows researchers to select different data collection approaches such as Participatory, Context-Triggered, or a Hybrid approach. It also provides additional features supporting researchers in collecting behavioral data. The first version of Minuku has showed its effectiveness in two published works, including the study described in Chapter 5. The improved version of Minuku will be utilized in three ongoing projects and three future projects.

7.2 Discussion

While each of the studies in Chapter 3, 4, and 5 provide a standalone set of design suggestions for contributing to specific research areas, this thesis attempts to take a more holistic view to treat the study results, taking them as a whole to provide more comprehensive implications for informing the design space in, and for informing the future research of context-aware application development. In the

following sections, I first discuss research challenges and propose new research questions in data capture. Then I envisage a C&P infrastructure integrating features in literature and features proposed in the chapters to achieve more effective use of C&P. Through illustrating the features in this infrastructures, I highlight new research as well as opportunities in creating this C&P infrastructure. Finally, I describe my current ongoing project and future work for this thesis.

7.2.1 Summary of Limitations

7.2.2 Research Challenges in Data Capture for Context-Awareness Development

Recently, Xiao (Xiao et al., 2013) reviewed a number of large-scale mobile crowdsensing projects and summarized three main barriers these research projects encountered. The first obstacle is heterogeneity of sensing hardware and mobile platforms that the mobile crowd uses for data collection tasks. The second obstacle is the burden of installing a separate proprietary application for every crowdsensing project in which users wish to participate. The third obstacle is the increasing network bandwidth demands of emerging crowdsensing applications, such as uploading multimedia data. The first obstacle can be partially addressed by Minuku because it currently only supports the Android system though within the Android system there still lacks a standardization of sensing hardware. The second obstacle can be tackled using only one platform, such as Minuku, for performing all data collection requests. The third obstacle, however, may need to be addressed by a new generation of networking technology. Compared to these obstacles that presumably can be resolved by more advanced and standardized mobile technology, in this section, I propose a set of new research questions that

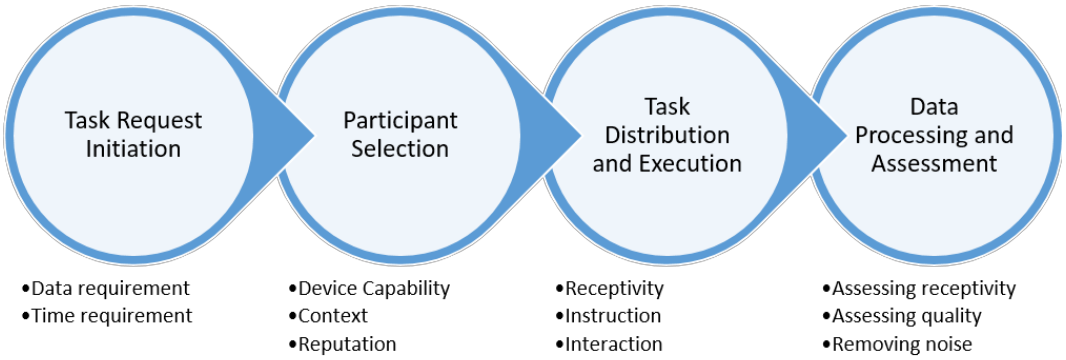


Figure 7.1 A presentation of data capture process via the mobile crowd.

need to be addressed for achieving more effective data capture. I highlight these research questions in illustrating a data capture process shown in Figure 7.1.

The first phase of data capture is task request initiation. According our experience in initiating a data collection request, data and time requirement are two optional but if present, decisive elements for making subsequent decisions about a capture plan. As we have discussed earlier in our reflections in Chapter 3, data requirement is largely based on the development activity in which data will be used. In the early stage of a design and development activity, it is likely that no specific and concrete data need has emerged, since it has not been clear what kind of data would be relevant for prototyping and evaluating the application. As a result, the development team is more likely to adopt a broad and opportunistic capture approach. As more specific needs of data emerge, the team may more often adopt a focused, specific, and targeted data capture compared to the early stage. However, one challenge for developers that we have shown earlier is the need for collecting highly specific type of data within a certain timeframe. As a result, this leads to a new research question:

RQ1: How do we support designers and developers in obtaining specific behavioral and contextual data, especially when the data request is time sensitive?

Based on the characteristic of the task, data requesters then need to determine who are suited for collecting the data. Suppose the team is sending a data request to the mobile crowd via a certain platform. Researchers have suggested a number of factors to be considered to improve the effectiveness of data collection, factors including whether the users' device is suitable for collecting the data needed (Das et al., 2010; Sasank Reddy, Samanta, et al., 2009); the cost for participants to complete the task given their current context (He et al., 2015; Konomi & Sasao, 2015; Sasank Reddy, Shilton, et al., 2009); whether the users can provide high quality of data based on their reputation (K. L. Huang et al., 2010a; Sasank Reddy et al., 2010; Truskinger et al., 2011) ; how responsive the users would be after sending them the request (Sasank Reddy et al., 2008); what the users' trustworthiness in their own social network is (Amintoosi & Kanhere, 2014b); and whether assigning the task to particular users increases the overall coverage of the data obtained or reduces the overall total cost for the request (Cardone et al., 2013; Sasank Reddy, Shilton, et al., 2009; Sasank Reddy et al., 2010; Singla & Krause, 2013). Given such a variety of factors to take into account, the challenge the team faces is choosing among these factors when selecting participants from the mobile crowd. For researchers, I highlight two researcher questions.

RQ2: What factors are more, and less indicative of users' performance and to the overall utility of the data obtained, respectively?

RQ3: How do we model an overall utility of a data collection task given the selected participants using the aforementioned factors?

The third phase is distributing the data collection task to selected participants. In this phase, one challenge is finding opportune moments to notify participants of the task. In recent years, numerous research attempts have been made to identify opportune moments to deliver notifications and questionnaires (J. E. Fischer, Greenhalgh, & Benford, 2011b; Pejovic & Musolesi, 2014d; Benjamin Poppinga et al., 2014; Sarker et al., 2014). However, none of these research efforts have investigated mobile users' receptivity to collecting behavioral and contextual data, which, may pose a different task switching cost—switch from the current activity to performing the requested activity, as opposed to answering a questionnaire. Although in Chapter 5 we have shown that mobile users' receptivity would differ according to the type of activity being collected, that study did not explore identifying opportune moments at breakpoints during an activity, which might lead to different results on mobile users' attentiveness and responsiveness to the task request, respectively. As a result, a research question remaining for this phase is:

RQ4: How do we model and find opportune moments for delivering data collection tasks requiring user interactions such as adding annotations?

After participants are willing to perform the task, it is important that the instruction for performing the task is specific and precisely described. Data requesters also need to determine whether they would employ a Participatory approach or an automated approach such as an Opportunistic Sensing approach or a Context-Triggered approach. They need to consider a number of factors for choosing a method that best suits their needs. The factors include but are not limited to: whether they would have time cleaning the noise of the data; whether they would prefer a longer but more complete data traces, or shorter but fragmented traces; whether they would like a broader data capture (e.g. using automated recording) or a narrowly focused capture that is potentially subject to a

self-selection bias (e.g. using manual recording); and how much burden and control they desire to give participants. Furthermore, they also need to decide whether any interaction with participants such as sending a prompt for increasing their awareness or reminding them for recording or annotation would be needed. The instruction then needs to be tailored according to the approach being employed. Although in Chapter 5 we have provided a pros and cons analysis of different approaches, it may not be realistic to anticipate that data requesters are well informed with the differences resulted by employing different approaches. As a result, researchers may need to play a more active role in informing data requesters during data capture, possibly providing a set of recommendations of the approaches, given the data requesters' needs. This phase involves, at least, two research questions:

RQ5: What would be a good way to inform data requesters about the pros and cons of different approaches?

RQ6: How do we automate the generation of approach and instruction recommendation based on the data requester's inputs regarding their data need?

In addition, we have shown that participants' recording and annotation behaviors are affected by the nature of the activity being collected. And their annotation timing can affect the characteristics of the annotation. However, it remains unclear to what extent these findings are generalizable to other types of activities in other contexts (e.g. eating activity in the indoor environment). Furthermore, although in Chapter 6 we propose providing specific instructions and reminders for data annotation, it is not clear whether this proposal actually leads to more detailed and higher quality of annotations. Thus, the question regarding to what extent the quality of annotations can be improved by a reminder or by more precise instructions remains.

RQ7: Do mobile users display similar recording and annotation behavioral patterns in collecting different types of activity? To what degree the features of different types of activity being collected impact mobile users' data collection behaviors?

RQ8: Does the provision of reminders for adding annotations and more precise instructions improve the quality of annotations made by mobile users?

Finally, after participants have collected data, data requesters would assess the data collected, and possibly also process the data if needed (e.g. cleaning noises). For each participant, data assessment may involve evaluating whether the data collected matches the specified data requirement, and the quality and quantity of data. This assessment can be converted into a score, taken as an input for a reputation framework to compute an overall reputation of participants. Data requesters can also assess the overall quality, quantity, and coverage of data upon each request. In this phase, I highlight three challenges in data assessment and reputation computation. First, previous research on computing participants' reputation based on the data assessment for mobile crowdsensing mostly focused on one specific type of data. While a generic mobile data collection such as Minuku is used whereby more diverse data types may be captured, it becomes a challenge to develop a metric for assessing data of different types. In addition, there is likely to be a suitability issue between participants and the task. That is, some participants are likely to be more suited to performing certain types of tasks than others. A new reputation system, thus, should consider possible suitability between participants and tasks when computing participants' reputation. Finally, some previous reputation frameworks also included participants' responsiveness (Sasank Reddy et al., 2008). It is, however, not clear how their notion of responsiveness was measured and operationalized. It is important to note that

receptivity is dependent on both attentiveness and the responsiveness to already attended tasks, as noted in Chapter 4. Both of the measures are likely to be influenced by when the task is received, with a different degree of influence. While it has not been clear which measure would be more crucial for a data requester to consider when selecting participants, a reputation framework considering participants' receptivity should be more cautious about punishing participants for being "unresponsive." After all, it is likely that a participant being unresponsive is because data requesters send the task at inappropriate moments. I argue that we need a more considerate reputation framework for addressing the aforementioned challenges. Thus, I propose two research questions for future research:

RQ9: How do we develop more accurate metric(s) for assessing the collected data

RQ10: How do we design a better participant reputation system in consideration of their *suitability and receptivity to requested tasks*

7.2.3 Towards a Comprehensive Capture-and-Playback Infrastructure

Having addressed the two research goals of this thesis, one long-term goal is to inform the design space of a comprehensive C&P platform that better supports both data capture and data use in context-aware applications development. Taking all the highlights from the chapters, I envisage an infrastructure supporting various activities involved in a C&P process, including data capture, data organization, and playback, respectively. Figure 7.2 presents the activities and relevant services involved in this infrastructure. The infrastructure has three important elements: Users (left), Platform (middle), and Devices (right). I also envision three major types of users on the platform: Participants (the mobile crowd), Data Requesters, and Data Users. Participants are those mobile users who

are assigned data collection tasks and contribute personal behavioral and contextual data. Data Requesters are those who send a data task request. Data Users are those who use the data for prototyping, testing, and evaluating context-aware applications. A person can play multiple roles when using the platform.

The envisaged flow for this infrastructure is as follows: 1) Data Requesters deploy a data collection project using the Management Portal of the platform, where the deployment includes a data collection request specifying a data and a time requirement, and a Minuku configuration. 2) A Task Manager interprets the task specification and passes the task information to a Participant Selection Service. 3) The Participant Selection Service selects suited participants based on a model that ranks participants according to their reputation obtained from a Reputation Framework, and according to the recent mobility and activity context of candidate participants. The Reputation Framework considers candidate participants' previous data contributions, recent receptivity, suitability to the specified task, and perhaps also their trustworthiness in their social network (Amintoosi & Kanhere, 2014a). 4) The Participant Selection Service makes a selection and distributes the task request to selected participants' devices on which Minuku is installed. 5) Minuku configures the device accordingly, including configuring which context sources to use for obtaining data. Minuku uses a receptivity model to find opportune moments to interact with the participants if specified in the task request. 6) Participants who have received and attended to the task request determine whether and when to perform the task, depending on the requirement, the instruction, his own availability, and the cost for performing the data collection task. 7) Collected data are transmitted to a designated database using some uploading policy such as a frequency for uploading data or only uploading data when Minuku is connected to a Wifi network. 8) Data Requesters receive notifications about incoming data contributions and use the Management Portal to manage and assess the quality of the contributions. The assessment becomes an

input of the Reputation Framework. Data requesters can annotate the collected data to give Data Users more information about the

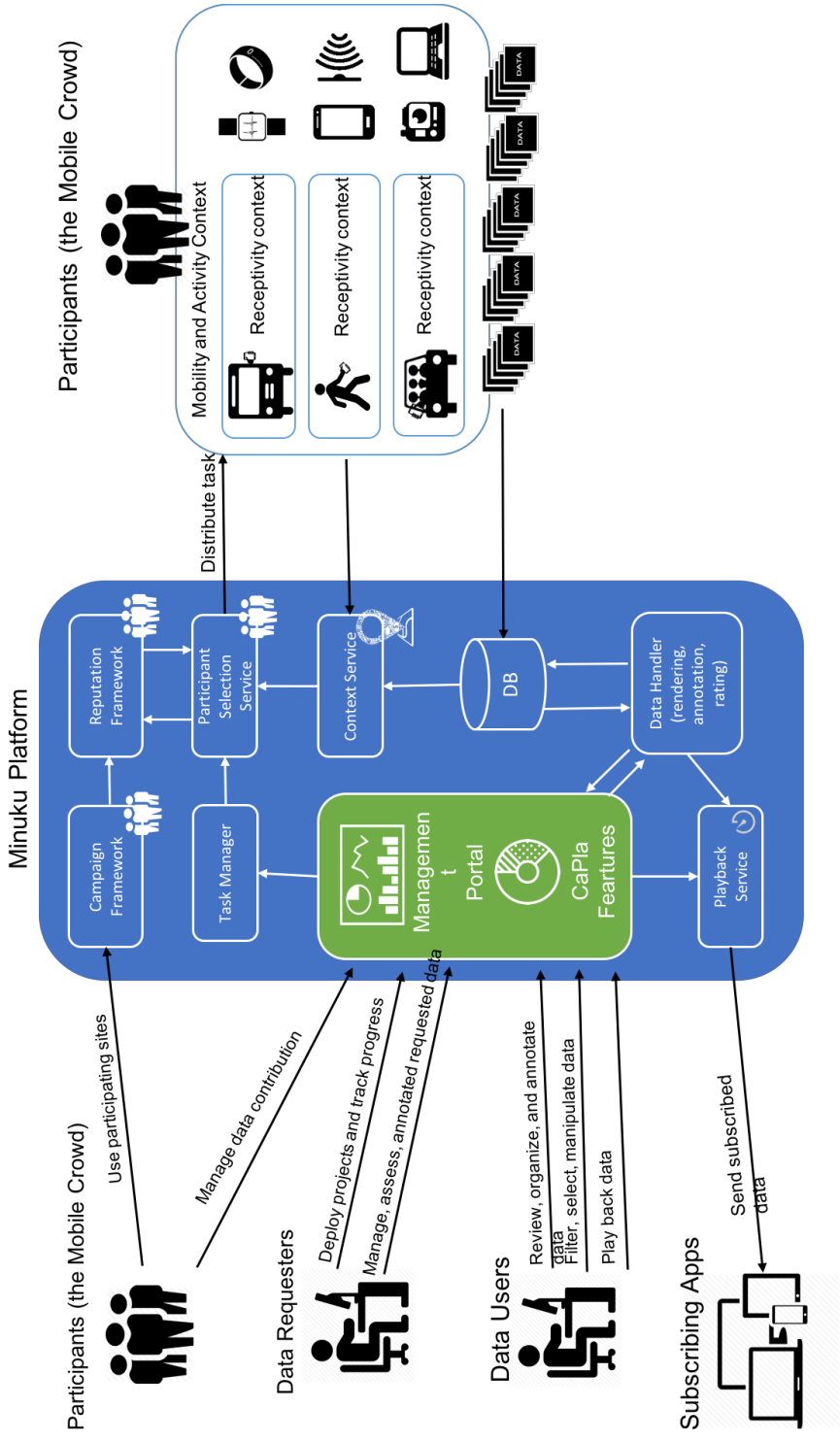


Figure 7.2 An envisaged Capture-and-Playback Infrastructure

data. 9) Data Users (if different from Data Requesters) use the Management Portal to review and organize the data they need for prototyping, testing, or evaluating their applications. They explore and filter data traces using a set of filters and via direct manipulation. They apply some renders that visualize the characteristics of the data traces. They add annotations to regions of traces they think useful for reuse or worth sharing with other Data Users. They may also see other Data Users' assessment, rating, and shared annotations on the data. 10) Data Users organize several sets of traces useful for different purposes and synthesize them as Episodes. They play back those Episodes to prototype, test, or evaluate their applications. A Playback Service periodically sends data tuples to the subscribing applications being tested and evaluated. 11) Data Users may modify some parts of data traces if the traces have not completely fulfilled their needs. This manipulation operation can be set to create a new copy of the data so that it will not modify the original copy. Finally, in addition to this 11-step process, participants can also manage their data contributions. They can also select a campaign to join, which can be on a social networking site that allows them to participate in data collection tasks.

This infrastructure integrates the proposed features in the chapters and the numerous features suggested to be useful in the literature. Although integrating these features seem to create a promising infrastructure that can hopefully provide a one ultimate solution for context-aware system development, combining them into one platform also leads to great complexity in the system design, thus making it challenging to design, to build, and to evaluate. Although previous research has provided some evidence showing benefits of each of the components (e.g. reputation framework, participant selection, task distribution) involved in the infrastructure, it is, however, unclear how these services would interplay with each other, what their weights are, and to what extent these services as a whole support context-aware application development. It is necessary then, that future

research develops new metrics to evaluate the integrated infrastructure (or a combination of some components of it), and to identify new challenges in using these components together. Furthermore, previous research efforts in mobile crowdsensing evaluate these components in a limited number of crowdsensing applications, all of which were collecting data of public phenomenon and information, instead of collecting individuals' behavioral and contextual data. As a result, future research will be needed to reexamine the effectiveness of these components in this new context.

Furthermore, I have been focusing on location and mobility data in the research works I describe in this thesis. Although mobility and location data is the most common type of contextual data used in commercial mobile applications to date, the advent of numerous Internet of Things (IoT) infrastructures[26,27,28,29] will bring a larger variety of context sources closer to interaction designers and developers. Sensors previously used more often in the research context have become more relevant and appealing to interaction designers and developers of commercial context-aware applications. Projects featured as smart-home or wearable technology have become widespread on crowdfunding sites such as Indiegogo[30], and KickStarter[31]. Given this emerging and ongoing trend, future research is

---

[26] http://googleresearch.blogspot.com/2016/02/announcing-google-internet-of-things.html

[27] http://www.apple.com/ios/homekit/

[28] http://www.ibm.com/internet-of-things/

[29] https://www.microsoft.com/en-us/server-cloud/internet-of-things/azure-iot-suite.aspx

[30] https://www.indiegogo.com/

[31] https://www.kickstarter.com

needed to examine the C&P features proposed in Chapter 3 for developing context-aware applications involving other, and multiple types of contextual data. New research questions include: what kinds of visualization, filter, and direction manipulation techniques can effectively support exploring, filtering, and selecting examples of different types of contextual data for playback? When heterogeneous types of contextual data are involved, what would be a good way to organize these data traces and present them to designers and developers in a more comprehensible way? How does a C&P platform incorporate emerging IoT infrastructures to capture heterogeneous contextual data from distributed devices that are not taken charged by a single participant? When it becomes common that mobile users have multiple devices for collecting data, what would be a good way to present these data to them to obtain annotations from these devices? I argue that future research is needed to address these research questions to keep the C&P platform timely and fulfill the emerging needs of designers and developers for using various types of context sources.

Finally, to ensure the sustainability of the C&P infrastructure, it is essential to investigate the motivational factors affecting participants' willingness of contributing personal behavioral and contextual data. Researchers have sought to enhance participants' engagement and level of participation in mobile crowdsensing (Omokaro, 2012), such as adding a gamification element (Sun, Zhu, Feng, & Yu, 2014). However, different motivational factors may have different weights in different crowdsensing projects, depending on the length of the project, the burden involved, the data being collected, and so forth. Compared to mobile crowdsensing, collecting personal behavioral and contextual data more often involves capturing personal sensitive and identifiable information, which is likely to have a larger impact on people's motivation in participating the research (Christin et al., 2011). In Chapter 5, I show that reducing user burden in controlling the tool and granting user control of the tool are two important factors

affecting participants' participation. However, it is unclear how large the impacts of these two factors have on participation in a long-term project. In addition, the findings are derived from participants' subjective self-reports. To what extent and how quickly their level of participation would decline over time is a remaining question. The study also did not attempt to explore a way to motivate participation. To ensure that designers and developers of context-aware applications can regularly and constantly obtain data contributions from the mobile crowd, it is crucial for future work to investigate the impact of different motivational factors in the context of collecting personal behavioral and contextual data.

7.2.4 Ongoing and Future Work

In the process of writing this thesis, I am also undertaking several projects as an ongoing and upcoming research works for this thesis. These projects mainly serve for continuing my exploration of the features useful for Minuku to capture behavioral and contextual data. All of these projects but one utilize Minuku for delivering questionnaires, including fixed-interval, context-triggered, and semi-randomized (randomizing sample times while ensuring a minimum interval between each prompt). In particular, one project is evaluating a new proposed data collection approach called Checkpoint-and-Remind (CAR), a hybrid approach combining a Participatory approach and a Context-Triggered In Situ approach to collect mobility data. The goal of the study is two folds. First, we seek to investigate whether the CAR approach outperforms both the Participatory approach and the Context-Triggered In Situ approach in collecting mobility data. Seconds, we also aim to evaluate the overall effectiveness of Minuku for capturing mobile users' mobility pattern in their daily lives. To capture the ground truth of participants' mobility history, we will also ask participants to wear a wearable camera, of which, the shooting rate is up to one photo per 10 seconds, a

significant improvement from the previous version of the camera. In addition, we will collect more types of contextual data and examine how the contextual information correlates with mobile users' receptivity to reminder notifications. Finally, we will also improve the reliability of the transportation detection on Minuku. We believe the results of the study will not only contribute to research involving capturing a mobility history of mobile users, but also demonstrate that our proposed Hybrid approach, informed by the results of Chapter 5, does improve the effectiveness of collecting annotated mobility data. The results will also be a demonstration of Minuku's capability of performing different data collection methods and performing situated actions.

In addition to these case studies for Minuku, in the short term, I aim to tackle the research challenges in investigating and improving mobile users' receptivity to collecting different types of annotated activity data in different contexts. I also aim to address the challenge of improving the quality of annotation made by the mobile crowd on activity data in the field. In the long term, my aim is address the research questions proposed in the previous two sections, with an ultimate goal to create a better infrastructure for supporting the development of context-aware applications.

7.3 Conclusion

In this thesis, I argue that an effective and efficient use of captured behavioral and contextual data for designing and developing context-aware applications is achievable through a combination of three components: 1) a capture-and-playback system facilitating prototyping, testing, and evaluation; 2) a set of good practices for effectively leveraging the mobile crowd to collect annotated behavioral and contextual data; and 3) a configurable, flexible, and extensible tool for collecting different types of behavioral and contextual data. Having addressed these conclusions, I highlight five contributions this thesis makes to the areas of

context-aware computing, mobile receptivity, and mobile crowdsensing/sourcing, respectively. The five conclusions are: 1) findings and lessons learned for informing the design space for supporting context-aware system development; 2) a capture-and-playback tool called CaPla that provides several features to support visualizing, filtering, selecting and modifying behavioral trace; 3) a better understanding of mobile users' interruption management practices on the phone and how ringer mode affects interruptibility and receptivity to incoming communication requests; 4) an understanding of the pros and cons of three different approaches to collecting annotated activity data through the mobile crowd in real word settings, and how nature of activities impact mobile users' data collection behavior; and finally 5) a configurable, flexible, and extensible mobile data collection tool Minuku that can monitor complex contextual conditions and schedule and perform highly situated actions.

Through making these contributions, my goal is to inform the design space for and to contribute to the research in context-aware applications development. To inform the design space, I provide an envisaged C&P infrastructure that integrates the proposed features in this thesis with the features suggested in the literature, with an assumption that the capture-and-playback approach is a promising direction to pursue. To make this infrastructure effective for supporting the C&P approach, I propose a number of research challenges and questions. It is my hope that future research efforts can address these challenges and questions to create a better infrastructure for designers, developers, and researchers to utilize to effectively capture behavioral and contextual traces, and to facilitate the design and development of context-aware applications.

BIBLIOGRAPHY

Aanensen, D. M., Huntley, D. M., Feil, E. J., al-Own, F. 'a, & Spratt, B. G. (2009). EpiCollect: Linking Smartphones to Web Applications for Epidemiology, Ecology and Community Data Collection. *PLoS ONE*, *4*(9), e6968. http://doi.org/10.1371/journal.pone.0006968

Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., & Steggles, P. (1999). Towards a better understanding of context and context-awareness. In *Handheld and ubiquitous computing* (pp. 304–307). Springer. Retrieved from http://link.springer.com.proxy.lib.umich.edu/chapter/10.1007/3-540-48157-5_29

Abowd, G. D., & Mynatt, E. D. (2000). Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *7*(1), 29–58.

Ackerman, M. S., Dong, T., Gifford, S., Kim, J., Newman, M. W., Prakash, A., … Dasgupta, P. (2009). Simplifying user-controlled privacy policies. *IEEE Pervasive Computing*, *8*, 28–32. http://doi.org/http://doi.ieeecomputersociety.org/10.1109/MPRV.2009.78

Agapie, E., Teevan, J., & Monroy-Hernández, A. (2015). Crowdsourcing in the Field: A Case Study Using Local Crowds for Event Reporting. In *Third AAAI Conference on Human Computation and Crowdsourcing*. Retrieved from http://www.aaai.org/ocs/index.php/HCOMP/HCOMP15/paper/view/11595

Ahmed, A., Yasumoto, K., Yamauchi, Y., & Ito, M. (2011). Distance and time based node selection for probabilistic coverage in people-centric sensing. In *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference on* (pp. 134–142). IEEE. Retrieved from http://ieeexplore.ieee.org.proxy.lib.umich.edu/xpls/abs_all.jsp?arnumber=5984884

Amintoosi, H., & Kanhere, S. S. (2014a). A Reputation Framework for Social Participatory Sensing Systems. *Mobile Networks and Applications*, *19*(1), 88–100. http://doi.org/10.1007/s11036-013-0455-x

Amintoosi, H., & Kanhere, S. S. (2014b). A Reputation Framework for Social Participatory Sensing Systems. *Mobile Networks and Applications*, *19*(1), 88–100. http://doi.org/10.1007/s11036-013-0455-x

Andrienko, G., Andrienko, N., Mladenov, M., Mock, M., & Pölitz, C. (2010). Discovering bits of place histories from people's activity traces. In *2010 IEEE Symposium on Visual Analytics Science and Technology (VAST)* (pp. 59–66). IEEE. http://doi.org/10.1109/VAST.2010.5652478

Arakawa, Y., & Matsuda, Y. (2016). Gamification Mechanism for Enhancing a Participatory Urban Sensing: Survey and Practical Results. *Journal of Information Processing*, *24*(1), 31–38. http://doi.org/10.2197/ipsjjip.24.31

Ashbrook, D., & Starner, T. (2003). Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, *7*(5), 275–286.

Auld, J., Williams, C., Mohammadian, A., & Nelson, P. (2009). An automated GPS-based prompted recall survey with learning algorithms. *Transportation Letters*, *1*(1), 59–79.

Avrahami, D., Fussell, S. R., & Hudson, S. E. (2008). IM waiting: timing and responsiveness in semi-synchronous communication. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work* (pp. 285–294). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1460610

Bachiller, R., Matthys, N., Cid, J. del, Joosen, W., Hughes, D., & Laerhoven, K. V. (2015). @migo: A Comprehensive Middleware Solution for Participatory Sensing Applications. In *2015 IEEE 14th International Symposium on Network Computing and Applications (NCA)* (pp. 1–8). http://doi.org/10.1109/NCA.2015.26

Baltrunas, L., Ludwig, B., Peer, S., & Ricci, F. (2011). Context-aware places of interest recommendations for mobile users. *Design, User Experience, and Usability*. *Theory, Methods, Tools and Practice*, 531–540.

Bao, L., & Intille, S. S. (2004a). Activity recognition from user-annotated acceleration data. In *Pervasive computing* (pp. 1–17). Springer. Retrieved from http://link.springer.com.proxy.lib.umich.edu/chapter/10.1007/978-3-540-24646-6_1

Bao, L., & Intille, S. S. (2004b). Activity Recognition from User-Annotated Acceleration Data. In A. Ferscha & F. Mattern (Eds.), *Pervasive Computing* (Vol. 3001, pp. 1–17). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://www.springerlink.com.proxy.lib.umich.edu/content/9aqflyk4f47khyjd/

Bardram, J. E. (2005). The Java Context Awareness Framework (JCAF)–a service infrastructure and programming framework for context-aware applications. *Pervasive Computing*, 98–115.

Barton, J. J., & Vijayaraghavan, V. (2003). *UBIWISE, a simulator for ubiquitous computing systems design* (Technical Report No. Tech. Report HPL-2003-93). Hewlett-Packard Labs.

Battestini, A., Setlur, V., & Sohn, T. (2010). A large scale study of text-messaging use. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services* (pp. 229–238). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1851638

Begole, J. "Bo," Matsakis, N. E., & Tang, J. C. (2004). Lilsys: Sensing Unavailability. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work* (pp. 511–514). New York, NY, USA: ACM. http://doi.org/10.1145/1031607.1031691

Belleflamme, P., Lambert, T., & Schwienbacher, A. (2014). Crowdfunding: Tapping the right crowd. *Journal of Business Venturing*, *29*(5), 585–609.

Bentley, F., & Metcalf, C. J. (2009). The Use of Mobile Social Presence. *IEEE Pervasive Computing*, *8*(4), 35–41. http://doi.org/10.1109/MPRV.2009.83

Beyer, H., & Holtzblatt, K. (1997). *Contextual Design : A Customer-Centered Approach to Systems Designs (Morgan Kaufmann Series in Interactive Technologies)*. {Morgan Kaufmann}. Retrieved from http://www.amazon.fr/exec/obidos/ASIN/1558604111/citeulike04-21

Bhattacharya, T., Kulik, L., & Bailey, J. (2012). Extracting significant places from mobile user GPS trajectories: a bearing change based approach. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems* (pp. 398–401). New York, NY, USA: ACM. http://doi.org/10.1145/2424321.2424374

Brouwers, N., & Langendoen, K. (2012). Pogo, a Middleware for Mobile Phone Sensing. In *Proceedings of the 13th International Middleware Conference*

(pp. 21–40). New York, NY, USA: Springer-Verlag New York, Inc. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=2442626.2442629

Burke, J. A., Estrin, D., Hansen, M., Parker, A., Ramanathan, N., Reddy, S., & Srivastava, M. B. (2006). Participatory sensing. *Center for Embedded Network Sensing*. Retrieved from http://escholarship.org/uc/item/19h777qd

Cao, X., Cong, G., & Jensen, C. S. (2010). Mining significant semantic locations from GPS data. *Proceedings of the VLDB Endowment*, *3*(1-2), 1009–1020.

Capatu, M., Regal, G., Schrammel, J., Mattheiss, E., Kramer, M., Batalas, N., & Tscheligi, M. (2014). Capturing mobile experiences: Context-and time-triggered in-situ questionnaires on a smartphone. *Measuring Behavior 2014*. Retrieved from http://dspace.library.uu.nl/bitstream/handle/1874/321355/Spink_etal2014_Proceedings_of_Measuring_Behavior_2014.pdf?sequence=1#page=76

Cardone, G., Foschini, L., Bellavista, P., Corradi, A., Borcea, C., Talasila, M., & Curtmola, R. (2013). Fostering participaction in smart cities: a geo-social crowdsensing platform. *Communications Magazine, IEEE*, *51*(6), 112–119.

Carter, S., Mankoff, J., & Heer, J. (2007). Momento: support for situated ubicomp experimentation. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 125–134). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1240644

Cerin, E., Szabo, A., & Williams, C. (2001). Is the Experience Sampling Method (ESM) appropriate for studying pre-competitive emotions? *Psychology of Sport and Exercise*, *2*(1), 27–45. http://doi.org/10.1016/S1469-0292(00)00009-1

Chang, Y., Hung, P., & Newman, M. W. (2012). TraceViz: “Brushing” for Location Based Services (p. Under Review). Presented at the Mobile HCI.

Chang, Y.-J., Hung, P.-Y., & Newman, M. (2012). TraceViz: “brushing” for location based services. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services companion* (pp. 219–220). New York, NY, USA: ACM. http://doi.org/10.1145/2371664.2371717

Chang, Y.-J., & Newman, M. W. (2012). Understanding How Trace Segmentation Impacts Transportation Mode Detection. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (pp. 625–626). New York, NY, USA: ACM. http://doi.org/10.1145/2370216.2370336

Chang, Y.-J., Paruthi, G., & Newman, M. W. (2015). A field study comparing approaches to collecting annotated activity data in real-world settings. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (pp. 671–682). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=2807524

Chang, Y.-J., & Tang, J. C. (2015). Investigating Mobile Users’ Ringer Mode Usage and Attentiveness and Responsiveness to Communication. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services* (pp. 6–15). New York, NY, USA: ACM. http://doi.org/10.1145/2785830.2785852

Chen, G., Kotz, D., & others. (2000). *A survey of context-aware mobile computing research*. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College. Retrieved from https://mmlab.snu.ac.kr/courses/2005_advanced_internet/handout/ppt/36%20-%20context_aware.pdf

Christensen, T. C., Barrett, L. F., Bliss-Moreau, E., Lebo, K., & Kaschub, C. (2003). A practical guide to experience-sampling procedures. *Journal of Happiness Studies*, *4*(1), 53–78.

Christin, D., Reinhardt, A., Kanhere, S. S., & Hollick, M. (2011). A survey on privacy in mobile participatory sensing applications. *Journal of Systems and Software*, *84*(11), 1928–1946.

Christin, D., Ro\s skopf, C., Hollick, M., Martucci, L. A., & Kanhere, S. S. (2013). Incognisense: An anonymity-preserving reputation framework for participatory sensing applications. *Pervasive and Mobile Computing*, *9*(3), 353–371.

Cleland, I., Han, M., Nugent, C., Lee, H., McClean, S., Zhang, S., & Lee, S. (2014). Evaluation of Prompted Annotation of Activity Data Recorded from a Smart Phone. *Sensors*, *14*(9), 15861–15879.

Cleland, I., Han, M., Nugent, C., Lee, H., Zhang, S., McClean, S., & Lee, S. (2013). Mobile Based Prompted Labeling of Large Scale Activity Data. In C. Nugent, A. Coronato, & J. Bravo (Eds.), *Ambient Assisted Living and Active Aging* (pp. 9–17). Springer International Publishing. Retrieved from http://link.springer.com.proxy.lib.umich.edu/chapter/10.1007/978-3-319-03092-0_2

Congleton, B., Ackerman, M. S., & Newman, M. W. (2008). The ProD framework for proactive displays. *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, 221–230. http://doi.org/10.1145/1449715.1449752

Consolvo, S., Arnstein, L., & Franza, B. (2002). User study techniques in the design and evaluation of a ubicomp environment. *UbiComp 2002: Ubiquitous Computing*, 281–290.

Consolvo, S., & Walker, M. (2003). Using the experience sampling method to evaluate ubicomp applications. *Pervasive Computing, IEEE*, *2*(2), 24–31.

Cooper, C. B., Dickinson, J., Phillips, T., & Bonney, R. (2007). Citizen science as a tool for conservation in residential ecosystems. *Ecology and Society*, *12*(2), 11.

Coric, V., & Gruteser, M. (2013). Crowdsensing maps of on-street parking spaces. In *Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference on* (pp. 115–122). IEEE. Retrieved from http://ieeexplore.ieee.org.proxy.lib.umich.edu/xpls/abs_all.jsp?arnumber=6569416

Crowley, C., Daniels, W., Bachiller, R., Joosen, W., & Hughes, D. (2014). Increasing user participation: An exploratory study of querying on the Facebook and Twitter platforms. *First Monday*, *19*(8). http://doi.org/10.5210/fm.v19i8.5325

Cruciani, F., Donnelly, M. P., Nugent, C. D., Parente, G., Paggetti, C., & Burns, W. (2011). DANTE: a video based annotation tool for smart environments. In *Sensor Systems and Software* (pp. 179–188). Springer. Retrieved from http://link.springer.com.proxy.lib.umich.edu/chapter/10.1007/978-3-642-23583-2_13

Csikszentmihalyi, M., & Larson, R. (1987). Validity and reliability of the experience-sampling method. *The Journal of Nervous and Mental Disease*, *175*(9), 526–536.

Dahlback, N., Jonsson, A., & Ahrenberg, L. (1993). Wizard of Oz studies: why and how. In *Proceedings of the 1st international conference on Intelligent user interfaces* (pp. 193–200). Orlando, Florida, United States: ACM. http://doi.org/10.1145/169891.169968

Danninger, M., Kluge, T., & Stiefelhagen, R. (2006a). MyConnector: analysis of context cues to predict human availability for communication. In *Proceedings of the 8th international conference on Multimodal interfaces* (pp. 12–19). Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1181001

Danninger, M., Kluge, T., & Stiefelhagen, R. (2006b). MyConnector: Analysis of Context Cues to Predict Human Availability for Communication. In *Proceedings of the 8th International Conference on Multimodal Interfaces* (pp. 12–19). New York, NY, USA: ACM. http://doi.org/10.1145/1180995.1181001

Das, T., Mohan, P., Padmanabhan, V. N., Ramjee, R., & Sharma, A. (2010). PRISM: platform for remote sensing using smartphones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services* (pp. 63–76). Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1814442

Davidoff, S., Lee, M. K., Dey, A. K., & Zimmerman, J. (2007). Rapidly exploring application design through speed dating. In *Proceedings of the 9th international conference on Ubiquitous computing* (pp. 429–446).

De Cristofaro, E., & Soriente, C. (2011). Short paper: PEPSI—privacy-enhanced participatory sensing infrastructure. In *Proceedings of the fourth ACM conference on Wireless network security* (pp. 23–28). Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1998418

De Guzman, E. S., Sharmin, M., & Bailey, B. P. (2007). Should I call now? Understanding what context is considered when deciding whether to initiate remote communication via mobile devices. In *Proceedings of Graphics interface 2007* (pp. 143–150). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1268542

Deng, L., & Cox, L. P. (2009). Livecompare: grocery bargain hunting through participatory sensing. In *Proceedings of the 10th workshop on Mobile Computing Systems and Applications* (p. 4). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1514415

DeVaul, R., & Dunn, S. (2001). *Real-Time Motion Classification for Wearable Computing Applications*.

Dey, A. K., Abowd, G. D., & Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Hum.-Comput. Interact.*, *16*(2), 97–166.

Dey, A. K., Hamid, R., Beckmann, C., Li, I., & Hsu, D. (2004). a CAPpella: programming by demonstration of context-aware applications. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 33–40). New York, NY, USA: ACM. http://doi.org/http://doi.acm.org.proxy.lib.umich.edu/10.1145/985692.985697

D'Hondt, E., Stevens, M., & Jacobs, A. (2013). Participatory noise mapping works! An evaluation of participatory sensing as an alternative to standard techniques for environmental monitoring. *Pervasive and Mobile Computing*, *9*(5), 681–694.

D'Hondt, E., Zaman, J., Philips, E., Boix, E. G., & De Meuter, W. (2014). Orchestration Support for Participatory Sensing Campaigns. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (pp. 727–738). New York, NY, USA: ACM. http://doi.org/10.1145/2632048.2632105

Dickerson, R. F., Gorlin, E. I., & Stankovic, J. A. (2011). Empath: A Continuous Remote Emotional Health Monitoring System for Depressive Illness. In *Proceedings of the 2Nd Conference on Wireless Health* (pp. 5:1–5:10). New York, NY, USA: ACM. http://doi.org/10.1145/2077546.2077552

Dingler, T., & Pielot, M. (2015). I'll be there for you: Quantifying Attentiveness towards Mobile Messaging. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services* (pp. 1–5). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=2785840

Dockray, S., Grant, N., Stone, A. A., Kahneman, D., Wardle, J., & Steptoe, A. (2010). A Comparison of Affect Ratings Obtained with Ecological Momentary Assessment and the Day Reconstruction Method. *Social Indicators Research*, *99*(2), 269–283. http://doi.org/10.1007/s11205-010-9578-7

Doherty, A., Kelly, P., & Foster, C. (2013). Wearable Cameras: Identifying Healthy Transportation Choices. *IEEE Pervasive Computing*, *12*(1), 44–47. http://doi.org/10.1109/MPRV.2013.21

Dong, Y. F., Kanhere, S., Chou, C. T., & Bulusu, N. (2008). Automatic collection of fuel prices from a network of mobile cameras. In *Distributed computing in sensor systems* (pp. 140–156). Springer. Retrieved from http://link.springer.com.proxy.lib.umich.edu/chapter/10.1007/978-3-540-69170-9_10

Dumais, S., Jeffries, R., Russell, D. M., Tang, D., & Teevan, J. (2014). Understanding User Behavior Through Log Data and Analysis. In J. S. Olson & W. A. Kellogg (Eds.), *Ways of Knowing in HCI* (pp. 349–372). Springer New York. Retrieved from http://link.springer.com.proxy.lib.umich.edu/chapter/10.1007/978-1-4939-0378-8_14

Eisenman, S. B., Lane, N. D., & Campbell, A. T. (2008). Techniques for improving opportunistic sensor networking performance. In *Distributed Computing in Sensor Systems* (pp. 157–175). Springer. Retrieved from

http://link.springer.com.proxy.lib.umich.edu/chapter/10.1007/978-3-540-69170-9_11

Falaki, H., Mahajan, R., & Estrin, D. (2011). Systemsens: a tool for monitoring usage in smartphone research deployments. In *Proceedings of the sixth international workshop on MobiArch* (pp. 25–30). Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1999923

Falaki, H., Mahajan, R., Kandula, S., Lymberopoulos, D., Govindan, R., & Estrin, D. (2010). Diversity in Smartphone Usage. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services* (pp. 179–194). New York, NY, USA: ACM. http://doi.org/10.1145/1814433.1814453

Farkas, K., Feher, G., Benczur, A., & Sidlo, C. (2015). Crowdsending based public transport information service in smart cities. *Communications Magazine, IEEE*, *53*(8), 158–165.

Ferreira, D., Gonçalves, J., Kostakos, V., Barkhuus, L., & Dey, A. K. (2014). Contextual Experience Sampling of Mobile Application Micro-Usage. Retrieved from http://www.ee.oulu.fi/~vassilis/files/papers/mobilehci14a.pdf

Ferris, B., Watkins, K., & Borning, A. (2010). OneBusAway: Results from providing real-time arrival information for public transit. In *Proceedings of the 28th international conference on Human factors in computing systems* (pp. 1807–1816).

Fetter, M., Seifert, J., & Gross, T. (2011a). Predicting Selective Availability for Instant Messaging. In P. Campos, N. Graham, J. Jorge, N. Nunes, P. Palanque, & M. Winckler (Eds.), *Human-Computer Interaction – INTERACT 2011* (pp. 503–520). Springer Berlin Heidelberg. Retrieved from http://link.springer.com.proxy.lib.umich.edu/chapter/10.1007/978-3-642-23765-2_35

Fetter, M., Seifert, J., & Gross, T. (2011b). Predicting Selective Availability for Instant Messaging. In P. Campos, N. Graham, J. Jorge, N. Nunes, P. Palanque, & M. Winckler (Eds.), *Human-Computer Interaction – INTERACT 2011* (pp. 503–520). Springer Berlin Heidelberg. Retrieved from http://link.springer.com.proxy.lib.umich.edu/chapter/10.1007/978-3-642-23765-2_35

Fischer, J. (2011). *Understanding receptivity to interruptions in mobile human-computer interaction*. University of Nottingham. Retrieved from http://etheses.nottingham.ac.uk/2499/

Fischer, J. E., Greenhalgh, C., & Benford, S. (2011a). Investigating episodes of mobile phone activity as indicators of opportune moments to deliver notifications. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services* (pp. 181–190). Retrieved from http://dl.acm.org/citation.cfm?id=2037402

Fischer, J. E., Greenhalgh, C., & Benford, S. (2011b). Investigating Episodes of Mobile Phone Activity As Indicators of Opportune Moments to Deliver Notifications. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services* (pp. 181–190). New York, NY, USA: ACM. http://doi.org/10.1145/2037373.2037402

Fischer, J. E., Yee, N., Bellotti, V., Good, N., Benford, S., & Greenhalgh, C. (2010). Effects of content and time of delivery on receptivity to mobile interruptions. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services* (pp. 103–112). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1851620

Fisher, R., & Simmons, R. (2011). Smartphone Interruptibility Using Density-Weighted Uncertainty Sampling with Reinforcement Learning. In *2011*

10th International Conference on Machine Learning and Applications and Workshops (ICMLA) (Vol. 1, pp. 436–441). http://doi.org/10.1109/ICMLA.2011.128

Fogarty, J., Hudson, S. E., Atkeson, C. G., Avrahami, D., Forlizzi, J., Kiesler, S., … Yang, J. (2005). Predicting human interruptibility with sensors. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *12*(1), 119–146.

Fogarty, J., Lai, J., & Christensen, J. (2004). Presence versus availability: the design and evaluation of a context-aware communication client. *International Journal of Human-Computer Studies*, *61*(3), 299–317.

Foremski, P., Gorawski, M., Grochla, K., & Polys, K. (2015). Energy-Efficient Crowdsensing of Human Mobility and Signal Levels in Cellular Networks. *Sensors*, *15*(9), 22060–22088. http://doi.org/10.3390/s150922060

Fouse, A., Weibel, N., Hutchins, E., & Hollan, J. D. (2011). ChronoViz: a system for supporting navigation of time-coded data. In *Proc*. *CHI 2011* (pp. 299–304). Vancouver, BC, Canada: ACM.

Fouse, A., Weibel, N., Hutchins, E., & Hollan, J. D. (2011). ChronoViz: a system for supporting navigation of time-coded data. In *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems* (pp. 299–304).

Froehlich, J., Chen, M. Y., Consolvo, S., Harrison, B., & Landay, J. A. (2007a). MyExperience: a system for in situ tracing and capturing of user feedback on mobile phones. In *Proc*. *MobiSys 2007* (pp. 57–70). http://doi.org/10.1145/1247660.1247670

Froehlich, J., Chen, M. Y., Consolvo, S., Harrison, B., & Landay, J. A. (2007b). MyExperience: a system for in situ tracing and capturing of user feedback on mobile phones. In *Proceedings of the 5th international conference on*

Mobile systems, applications and services (pp. 57–70). Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1247670

Froehlich, J., Chen, M. Y., Consolvo, S., Harrison, B., & Landay, J. A. (2007c). MyExperience: a system for in situ tracing and capturing of user feedback on mobile phones. In *Proceedings of the 5th international conference on Mobile systems, applications and services* (pp. 57–70). Retrieved from http://dl.acm.org/citation.cfm?id=1247670

Froehlich, J., Dillahunt, T., Klasnja, P., Mankoff, J., Consolvo, S., Harrison, B., & Landay, J. A. (2009). UbiGreen: investigating a mobile tool for tracking and supporting green transportation habits. In *Proceedings of the 27th international conference on Human factors in computing systems* (pp. 1043–1052). New York, NY, USA: ACM. http://doi.org/10.1145/1518701.1518861

Gaggioli, A., Pioggia, G., Tartarisco, G., Baldus, G., Corda, D., Cipresso, P., & Riva, G. (2013). A mobile data collection platform for mental health research. *Personal and Ubiquitous Computing*, *17*(2), 241–251. http://doi.org/10.1007/s00779-011-0465-2

Ganti, R. K., Pham, N., Tsai, Y.-E., & Abdelzaher, T. F. (2008). PoolView: stream privacy for grassroots participatory sensing. In *Proceedings of the 6th ACM conference on Embedded network sensor systems* (pp. 281–294). Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1460440

Ganti, R. K., Ye, F., & Lei, H. (2011). Mobile crowdsensing: Current state and future challenges. *Communications Magazine, IEEE*, *49*(11), 32–39.

Gao, H., Liu, C. H., Wang, W., Zhao, J., Song, Z., Su, X., … Leung, K. K. (2015). A Survey of Incentive Mechanisms for Participatory Sensing. *Communications Surveys & Tutorials, IEEE*, *17*(2), 918–943.

Gaonkar, S., Li, J., Choudhury, R. R., Cox, L., & Schmidt, A. (2008). Micro-Blog: Sharing and Querying Content Through Mobile Phones and Social Participation. In *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services* (pp. 174–186). New York, NY, USA: ACM. http://doi.org/10.1145/1378600.1378620

Gerken, J., Dierdorf, S., Schmid, P., Sautner, A., & Reiterer, H. (2010). Pocket Bee: A Multi-modal Diary for Field Research. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries* (pp. 651–654). New York, NY, USA: ACM. http://doi.org/10.1145/1868914.1868996

Goncalves, J., Hosio, S., Ferreira, D., & Kostakos, V. (2014). Game of Words: Tagging Places Through Crowdsourcing on Public Displays. In *Proceedings of the 2014 Conference on Designing Interactive Systems* (pp. 705–714). New York, NY, USA: ACM. http://doi.org/10.1145/2598510.2598514

Grandhi, S. A., Laws, N., Amento, B., & Jones, Q. (2008). The Importance of“ who” and“ what” in Interruption Management: Empirical Evidence from a Cell Phone Use Study. *AMCIS 2008 Proceedings*, 79.

Greenhalgh, C., French, A., Tennent, P., Humble, J., & Crabtree, A. (2007). From replaytool to digital replay system. In *3rd International Conference on e-Social Science*.

Hachem, S., Pathak, A., & Issarny, V. (2013). Probabilistic registration for large-scale mobile participatory sensing. In *Pervasive Computing and Communications (PerCom), 2013 IEEE International Conference on* (pp. 132–140). IEEE. Retrieved from http://ieeexplore.ieee.org.proxy.lib.umich.edu/xpls/abs_all.jsp?arnumber=6526723

Harada, S., Lester, J., Patel, K., Saponas, T. S., Fogarty, J., Landay, J. A., & Wobbrock, J. O. (2008). VoiceLabel: using speech to label mobile sensor data. In *Proceedings of the 10th international conference on Multimodal interfaces* (pp. 69–76). New York, NY, USA: ACM. http://doi.org/10.1145/1452392.1452407

Hartmann, B., Abdulla, L., Mittal, M., & Klemmer, S. R. (2007). Authoring sensor-based interactions by demonstration with direct manipulation and pattern recognition. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 145–154). San Jose, California, USA: ACM. http://doi.org/10.1145/1240624.1240646

Hartmann, B., Klemmer, S. R., Bernstein, M., Abdulla, L., Burr, B., Robinson-Mosher, A., & Gee, J. (2006). Reflective physical prototyping through integrated design, test, and analysis. In *Proceedings of the 19th annual ACM symposium on User interface software and technology* (pp. 299–308). Montreux, Switzerland: ACM. http://doi.org/10.1145/1166253.1166300

Hashemian, M., Knowles, D., Calver, J., Qian, W., Bullock, M. C., Bell, S., … Stanley, K. G. (2012). iEpi: An End to End Solution for Collecting, Conditioning and Utilizing Epidemiologically Relevant Data. In *Proceedings of the 2Nd ACM International Workshop on Pervasive Wireless Healthcare* (pp. 3–8). New York, NY, USA: ACM. http://doi.org/10.1145/2248341.2248345

Heimerl, K., Gawalt, B., Chen, K., Parikh, T., & Hartmann, B. (2012). CommunitySourcing: engaging local crowds to perform expert work via physical kiosks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1539–1548). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=2208619

He, Z., Cao, J., & Liu, X. (2015). High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility. In *Computer Communications (INFOCOM), 2015 IEEE Conference on* (pp. 2542–2550). IEEE. Retrieved from http://ieeexplore.ieee.org.proxy.lib.umich.edu/xpls/abs_all.jsp?arnumber=7218644

Hicks, J., Ramanathan, N., Kim, D., Monibi, M., Selsky, J., Hansen, M., & Estrin, D. (2010). Andwellness: an open mobile system for activity and experience sampling. In *Wireless health 2010* (pp. 34–43). Retrieved from http://dl.acm.org/citation.cfm?id=1921087

Ho, J., & Intille, S. S. (2005). Using context-aware computing to reduce the perceived burden of interruptions from mobile devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 909–918). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1055100

Hong, J. I., & Landay, J. A. (2004). An architecture for privacy-sensitive ubiquitous computing. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services* (pp. 177–189).

Horvitz, E., Koch, P., Kadie, C. M., & Jacobs, A. (2002). Coordinate: Probabilistic forecasting of presence and availability. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence* (pp. 224–233). Morgan Kaufmann Publishers Inc. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=2073903

Hosio, S., Goncalves, J., Lehdonvirta, V., Ferreira, D., & Kostakos, V. (2014). Situated Crowdsourcing Using a Market Model. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (pp. 55–64). New York, NY, USA: ACM. http://doi.org/10.1145/2642918.2647362

Huang, C.-C. J., Yang, R., & Newman, M. W. (2015). The potential and challenges of inferring thermal comfort at home using commodity sensors. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (pp. 1089–1100). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=2805831

Huang, K. L., Kanhere, S. S., & Hu, W. (2010a). Are you contributing trustworthy data?: the case for a reputation system in participatory sensing. In *Proceedings of the 13th ACM international conference on Modeling, analysis, and simulation of wireless and mobile systems* (pp. 14–22). Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1868526

Huang, K. L., Kanhere, S. S., & Hu, W. (2010b). Are you contributing trustworthy data?: the case for a reputation system in participatory sensing. In *Proceedings of the 13th ACM international conference on Modeling, analysis, and simulation of wireless and mobile systems* (pp. 14–22). Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1868526

Hufford, M. R. (2007). Special methodological challenges and opportunities in ecological momentary assessment. *The Science of Real-Time Data Capture: Self-Reports in Health Research*, 54–75.

Hufford, M. R., Shiffman, S., Paty, J., & Stone, A. A. (2001). Ecological Momentary Assessment: Real-world, real-time measurement of patient experience. Retrieved from http://psycnet.apa.org.proxy.lib.umich.edu/psycinfo/2001-05276-004

Hu, K., Wang, Y., Rahman, A., & Sivaraman, V. (2014). Personalising pollution exposure estimates using wearable activity sensors. In *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and*

Information Processing (ISSNIP) (pp. 1–6). http://doi.org/10.1109/ISSNIP.2014.6827617

Hunter, R., Donnelly, M. P., Finlay, D. D., Moore, G., & Booth, N. (2013). Capture and Access Tools for Event Annotation and Visualisation. In *UBICOMM 2013, The Seventh International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies* (pp. 134–139). Retrieved from http://www.thinkmind.org/index.php?view=article&articleid=ubicomm_2013_7_30_10126

Hu, X., Liu, Q., Zhu, C., Leung, V., Chu, T. H., & Chan, H. C. (2013). A mobile crowdsensing system enhanced by cloud-based social networking services. In *Proceedings of the First International Workshop on Middleware for Cloud-enabled Sensing* (p. 3). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=2541604

Ilarri, S., Wolfson, O., & Delot, T. (2014). Collaborative sensing for urban transportation. *IEEE Data Engineering Bulletin*, *37*(4), 3–14.

Inria. (n.d.). CrowdSignals.io. Retrieved March 3, 2016, from http://crowdsignals.io/

Intille, S. S. (2007). Technological innovations enabling automatic, context-sensitive ecological momentary assessment. *The Science of Real-Time Data Capture*. *Self-Reports in Health Research*, 308–337.

Intille, S. S., Rondoni, J., Kukla, C., Ancona, I., & Bao, L. (2003). A context-aware experience sampling tool. In *CHI'03 extended abstracts on Human factors in computing systems* (pp. 972–973). Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=766101

Iqbal, S. T., & Bailey, B. P. (2008). Effects of intelligent notification management on users and their tasks. In *Proceedings of the SIGCHI Conference on*

Human Factors in Computing Systems (pp. 93–102). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1357070

Izadi, S., Brignull, H., Rodden, T., Rogers, Y., & Underwood, M. (2003). Dynamo: a public interactive surface supporting the cooperative sharing and exchange of media. In *Proceedings of the 16th annual ACM symposium on User interface software and technology* (pp. 159–168).

Jaimes, L. G., Vergara-Laurens, I. J., & Raij, A. (2015). A Survey of Incentive Techniques for Mobile Crowd Sensing. *Internet of Things Journal, IEEE*, *2*(5), 370–380.

Jayaraman, P. P., Perera, C., Georgakopoulos, D., & Zaslavsky, A. (2013). Efficient opportunistic sensing using mobile collaborative platform mosden. In *Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 2013 9th International Conference Conference on* (pp. 77–86). IEEE. Retrieved from http://ieeexplore.ieee.org.proxy.lib.umich.edu/xpls/abs_all.jsp?arnumber=6679972

Joki, A., Burke, J. A., & Estrin, D. (2007). Campaignr: a framework for participatory data collection on mobile phones. Retrieved from http://escholarship.org/uc/item/8v01m8wj.pdf

Kahneman, D., Krueger, A. B., Schkade, D. A., Schwarz, N., & Stone, A. A. (2004). A survey method for characterizing daily life experience: The day reconstruction method. *Science*, *306*(5702), 1776–1780.

Kanhere, S. S. (2011). Participatory Sensing: Crowdsourcing Data from Mobile Smartphones in Urban Spaces. In *Mobile Data Management (MDM), 2011 12th IEEE International Conference on* (Vol. 2, pp. 3–6).

Kanjo, E. (2010). Noisespy: A real-time mobile phone platform for urban noise monitoring and mapping. *Mobile Networks and Applications*, *15*(4), 562–574.

Kanjo, E., Bacon, J., Roberts, D., & Landshoff, P. (2009). MobSens: Making smart phones smarter. *Pervasive Computing, IEEE*, *8*(4), 50–57.

Kansal, A., Nath, S., Liu, J., & Zhao, F. (2007). Senseweb: An infrastructure for shared sensing. *IEEE Multimedia*, (4), 8–13.

Kantarci, B., & Mouftah, H. T. (2014). Reputation-based Sensing-as-a-Service for crowd management over the cloud. In *Communications (ICC), 2014 IEEE International Conference on* (pp. 3614–3619). IEEE. Retrieved from http://ieeexplore.ieee.org.proxy.lib.umich.edu/xpls/abs_all.jsp?arnumber=6883882

Kapadia, A., Kotz, D., & Triandopoulos, N. (2009). Opportunistic sensing: Security challenges for the new paradigm. In *Communication Systems and Networks and Workshops, 2009. COMSNETS 2009. First International* (pp. 1–10). IEEE. Retrieved from http://ieeexplore.ieee.org.proxy.lib.umich.edu/xpls/abs_all.jsp?arnumber=4808850

Kelly, P., Doherty, A., Mizdrak, A., Marshall, S., Kerr, J., Legge, A., … Foster, C. (2014). High group level validity but high random error of a self-report travel diary, as assessed by wearable cameras. *Journal of Transport & Health*, *1*(3), 190–201. http://doi.org/10.1016/j.jth.2014.04.003

Khan, W. Z., Xiang, Y., Aalsalem, M. Y., & Arshad, Q. (2013). Mobile Phone Sensing Systems: A Survey. *IEEE Communications Surveys Tutorials*, *15*(1), 402–427. http://doi.org/10.1109/SURV.2012.031412.00077

Kim, J., Kikuchi, H., & Yamamoto, Y. (2013). Systematic comparison between ecological momentary assessment and day reconstruction method for

fatigue and mood states in healthy adults. *British Journal of Health Psychology*, *18*(1), 155–167. http://doi.org/10.1111/bjhp.12000

Kim, S., Mankoff, J., & Paulos, E. (2013). Sensr: Evaluating a Flexible Framework for Authoring Mobile Data-collection Tools for Citizen Science. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work* (pp. 1453–1462). New York, NY, USA: ACM. http://doi.org/10.1145/2441776.2441940

Klumb, P. L., & Baltes, M. M. (1999). Validity of retrospective time-use reports in old age. *Applied Cognitive Psychology*, *13*(6), 527–539.

Konomi, S. 'ichi, & Sasao, T. (2015). The Use of Colocation and Flow Networks in Mobile Crowdsourcing. In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers* (pp. 1343–1348). New York, NY, USA: ACM. http://doi.org/10.1145/2800835.2800967

Kotsiantis, S. B. (2007). *Supervised machine learning: A review of classification techniques*. Retrieved from https://books.google.com/books?hl=en&lr=&id=vLiTXDHr_sYC&oi=fnd&pg=PA3&ots=CXpyyzYEfq&sig=-U7j6sg0qBZ1-ZW_2up6JQ3lOBI

Kwapisz, J. R., Weiss, G. M., & Moore, S. A. (2011). Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, *12*(2), 74–82.

Lane, N. D., Chon, Y., Zhou, L., Zhang, Y., Li, F., Kim, D., … Cha, H. (2013). Piggyback CrowdSensing (PCS): Energy Efficient Crowdsourcing of Mobile Sensor Data by Exploiting Smartphone App Opportunities. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems* (pp. 7:1–7:14). New York, NY, USA: ACM. http://doi.org/10.1145/2517351.2517372

Lane, N. D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., & Campbell, A. T. (2010). A survey of mobile phone sensing. *Communications Magazine, IEEE*, *48*(9), 140–150.

Lane, N. D., Mohammod, M., Lin, M., Yang, X., Lu, H., Ali, S., … Campbell, A. (2011). BeWell: A smartphone application to monitor, model and promote wellbeing. In *5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth2011)*. Retrieved from http://www.cs.dartmouth.edu/~tanzeem/pubs/PervasiveHealth_BeWell.pdf

Lane, N. D., Xu, Y., Lu, H., Hu, S., Choudhury, T., Campbell, A. T., & Zhao, F. (2011). Enabling large-scale human activity inference on smartphones using community similarity networks (csn). In *Proceedings of the 13th international conference on Ubiquitous computing* (pp. 355–364). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=2030160

Lasecki, W. S., Song, Y. C., Kautz, H., & Bigham, J. P. (2013). Real-time crowd labeling for deployable activity recognition. In *Proceedings of the 2013 conference on Computer supported cooperative work* (pp. 1203–1212). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=2441912

Lathia, N., Rachuri, K., Mascolo, C., & Roussos, G. (2013). Open Source Smartphone Libraries for Computational Social Science. In *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication* (pp. 911–920). New York, NY, USA: ACM. http://doi.org/10.1145/2494091.2497345

Liao, L., Fox, D., & Kautz, H. (2007). Extracting places and activities from gps traces using hierarchical conditional random fields. *The International Journal of Robotics Research*, *26*(1), 119–134.

Lin, K., Kansal, A., Lymberopoulos, D., & Zhao, F. (2010). Energy-accuracy trade-off for continuous mobile device location. In *Proceedings of the 8th international conference on Mobile systems, applications, and services* (pp. 285–298). Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1814462

Linnap, M., & Rice, A. (2014). Managed Participatory Sensing with YouSense. *Journal of Urban Technology*, *21*(2), 9–26. http://doi.org/10.1080/10630732.2014.888216

Li, Y., Hong, J. I., & Landay, J. A. (2004). Topiary: a tool for prototyping location-enhanced applications. In *Proceedings of the 17th annual ACM symposium on User interface software and technology* (pp. 217–226). ACM.

Li, Y., & Landay, J. A. (2008). Activity-based prototyping of ubicomp applications for long-lived, everyday human activities. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems* (pp. 1303–1312). New York, NY, USA: ACM. http://doi.org/http://doi.acm.org.proxy.lib.umich.edu/10.1145/1357054.1357259

Ljungstrand, P. (2001). Context awareness and mobile phones. *Personal and Ubiquitous Computing*, *5*(1), 58–61.

Lu, H., Pan, W., Lane, N. D., Choudhury, T., & Campbell, A. T. (2009). SoundSense: Scalable Sound Sensing for People-centric Applications on Mobile Phones. In *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services* (pp. 165–178). New York, NY, USA: ACM. http://doi.org/10.1145/1555816.1555834

Lu, H., Yang, J., Liu, Z., Lane, N. D., Choudhury, T., & Campbell, A. T. (2010). The Jigsaw Continuous Sensing Engine for Mobile Phone Applications. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor*

Systems (pp. 71–84). New York, NY, USA: ACM. http://doi.org/10.1145/1869983.1869992

MacIntyre, B., Gandy, M., Dow, S., & Bolter, J. D. (2004). DART: a toolkit for rapid design exploration of augmented reality experiences. In *Proc*. *UIST 2010* (pp. 197–206). http://doi.org/10.1145/1029632.1029669

Maisonneuve, N., Stevens, M., Niessen, M. E., Hanappe, P., & Steels, L. (2009). Citizen noise pollution monitoring. In *Proceedings of the 10th Annual International Conference on Digital Government Research: Social Networks: Making Connections between Citizens, Data and Government* (pp. 96–103). Digital Government Society of North America. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1556176.1556198

Marmasse, N., & Schmandt, C. (2002). A user-centered location model. *Personal and Ubiquitous Computing*, *6*(5-6), 318–321.

McCarthy, J. F., Congleton, B., & Harper, F. M. (n.d.). Sharing Online Photos via Proactive Displays in the Physical Workplace.

McFarlane, D. C., & Latorella, K. A. (2002). The scope and importance of human interruption in human-computer interaction design. *Human-Computer Interaction*, *17*(1), 1–61.

Meschtscherjakov, A., Reitberger, W., & Tscheligi, M. (2010). MAESTRO: orchestrating user behavior driven and context triggered experience sampling. In *Proceedings of the 7th International Conference on Methods and Techniques in Behavioral Research* (p. 29). Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1931373

Mihalic, K., & Tscheligi, M. (2007). “Divert: mother-in-law”: representing and evaluating social context on mobile devices. In *Proceedings of the 9th international conference on Human computer interaction with mobile*

devices and services (pp. 257–264). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1378016

Min, J.-K., Wiese, J., Hong, J. I., & Zimmerman, J. (2013). Mining smartphone data to classify life-facets of social relationships. In *Proceedings of the 2013 conference on Computer supported cooperative work* (pp. 285–294). Retrieved from http://dl.acm.org/citation.cfm?id=2441810

Mohan, P., Padmanabhan, V. N., & Ramjee, R. (2008). Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *Proceedings of the 6th ACM conference on Embedded network sensor systems* (pp. 323–336). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1460444

Morren, M., Dulmen, S., Ouwerkerk, J., & Bensing, J. (2009). Compliance with momentary pain measurement using electronic diaries: a systematic review. *European Journal of Pain*, *13*(4), 354–365.

Mulder, I., Ter Hofte, G. H., & Kort, J. (2005). SocioXensor: Measuring user behaviour and user eXperience in conteXt with mobile devices. In *Proceedings of Measuring Behavior* (pp. 355–358). Retrieved from https://www.researchgate.net/profile/Ingrid_Mulder2/publication/238742075_SocioXensor_Measuring_user_behaviour_and_user_eXperience_in_conteXt_with_mobile_devices/links/00b49527fa728900e8000000.pdf

Mun, M., Reddy, S., Shilton, K., Yau, N., Burke, J., Estrin, D., … Boda, P. (2009). PEIR, the personal environmental impact report, as a platform for participatory sensing systems research. In *Proceedings of the 7th international conference on Mobile systems, applications, and services* (pp. 55–68). New York, NY, USA: ACM. http://doi.org/10.1145/1555816.1555823

Musolesi, M., Piraccini, M., Fodor, K., Corradi, A., & Campbell, A. T. (2010). Supporting Energy-Efficient Uploading Strategies for Continuous Sensing

Applications on Mobile Phones. In P. Floréen, A. Krüger, & M. Spasojevic (Eds.), *Pervasive Computing* (pp. 355–372). Springer Berlin Heidelberg. Retrieved from http://link.springer.com.proxy.lib.umich.edu/chapter/10.1007/978-3-642-12654-3_21

Nagel, K. S., Hudson, J. M., & Abowd, G. D. (2004). Predictors of availability in home life context-mediated communication. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work* (pp. 497–506). New York, NY, USA: ACM. http://doi.org/10.1145/1031607.1031689

Nazneen, N., Rozga, A., Romero, M., Findley, A. J., Call, N. A., Abowd, G. D., & Arriaga, R. I. (2012). Supporting parents for in-home capture of problem behaviors of children with developmental disabilities. *Personal and Ubiquitous Computing*, *16*(2), 193–207.

Newman, M. W., Ackerman, M. S., Kim, J., Prakash, A., Hong, Z., Mandel, J., & Dong, T. (2010). Bringing the field into the lab. In *Proceedings of the Proc. UIST 2010* (p. 105). New York, New York, USA. http://doi.org/10.1145/1866029.1866048

Nextbus. (n.d.). Retrieved from http://www.nextbus.com/homepage/

Nowak, S., & Rüger, S. (2010). How Reliable Are Annotations via Crowdsourcing: A Study About Inter-annotator Agreement for Multi-label Image Annotation. In *Proceedings of the International Conference on Multimedia Information Retrieval* (pp. 557–566). New York, NY, USA: ACM. http://doi.org/10.1145/1743384.1743478

Omokaro, O. (2012). A Framework to Promote User Engagement in Participatory Sensing Applications. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (pp. 548–551). New York, NY, USA: ACM. http://doi.org/10.1145/2370216.2370306

O'Neill, E., Klepal, M., Lewis, D., O'Donnell, T., O'Sullivan, D., & Pesch, D. (2005). A testbed for evaluating human interaction with ubiquitous computing environments. In *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2005. Tridentcom 2005. First International Conference on* (pp. 60–69). IEEE.

Paek, J., Kim, J., & Govindan, R. (2010). Energy-efficient Rate-adaptive GPS-based Positioning for Smartphones. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services* (pp. 299–314). New York, NY, USA: ACM. http://doi.org/10.1145/1814433.1814463

Paek, J., Kim, K.-H., Singh, J. P., & Govindan, R. (2011). Energy-efficient positioning for smartphones using cell-id sequence matching. In *Proceedings of the 9th international conference on Mobile systems, applications, and services* (pp. 293–306). Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=2000024

Palmblad, M., & Tiplady, B. (2004). Electronic diaries and questionnaires: Designing user interfaces that are easy for all patients to use. *Quality of Life Research*, *13*(7), 1199–1207. http://doi.org/10.1023/B:QURE.0000037501.92374.e1

Patel, K., Bancroft, N., Drucker, S. M., Fogarty, J., Ko, A. J., & Landay, J. (2010). Gestalt: integrated support for implementation and analysis in machine learning. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology* (pp. 37–46). New York, NY, USA: ACM. http://doi.org/10.1145/1866029.1866038

Patel, S. N., Kientz, J. A., Hayes, G. R., Bhat, S., & Abowd, G. D. (2006). Farther than you may think: An empirical investigation of the proximity of users to their mobile phones. In *UbiComp 2006: Ubiquitous Computing* (pp.

123–140). Springer. Retrieved from http://link.springer.com.proxy.lib.umich.edu/chapter/10.1007/11853565_8

Paulos, E., Honicky, R. J., & Goodman, E. (2007). Sensing atmosphere. *Human-Computer Interaction Institute*, 203.

Paxton, M., & Benford, S. (2009). Experiences of participatory sensing in the wild. In *Proceedings of the 11th international conference on Ubiquitous computing* (pp. 265–274). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1620584

Pejovic, V., & Musolesi, M. (2014a). Anticipatory Mobile Computing for Behaviour Change Interventions. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication* (pp. 1025–1034). New York, NY, USA: ACM. http://doi.org/10.1145/2638728.2641284

Pejovic, V., & Musolesi, M. (2014b). InterruptMe: designing intelligent prompting mechanisms for pervasive applications. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (pp. 897–908). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=2632062

Pejovic, V., & Musolesi, M. (2014c). InterruptMe: designing intelligent prompting mechanisms for pervasive applications. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (pp. 897–908). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=2632062

Pejovic, V., & Musolesi, M. (2014d). InterruptMe: Designing Intelligent Prompting Mechanisms for Pervasive Applications. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (pp. 897–908). New York, NY, USA: ACM. http://doi.org/10.1145/2632048.2632062

Pielot, M. (2014). Large-scale evaluation of call-availability prediction. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (pp. 933–937). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=2632060

Pielot, M., Church, K., & de Oliveira, R. (2014). An In-Situ Study of Mobile Phone Notifications. In *Proc. MobileHCI* (Vol. 14). Retrieved from http://pielot.org/pubs/Pielot2014-MobileHCI-Notifications.pdf

Pielot, M., de Oliveira, R., Kwak, H., & Oliver, N. (2014a). Didn't you see my message?: predicting attentiveness to mobile instant messages. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems* (pp. 3319–3328). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=2556973

Pielot, M., de Oliveira, R., Kwak, H., & Oliver, N. (2014b). Didn't you see my message? predicting attentiveness to mobile instant messages. In *Proc. CHI*. Retrieved from http://pielot.org/pubs/Pielot2014-CHI-AttPred.pdf

Pielot, M., Dingler, T., San Pedro, J., & Oliver, N. (2015). When Attention is not Scarce-Detecting Boredom from Mobile Phone Usage. In *Proc. of UbiComp*. Retrieved from http://pielot.org/pubs/Pielot2015-UbiComp-Boredom-Detection.pdf

Poppinga, B., Heuten, W., & Boll, S. (2014). Sensor-Based Identification of Opportune Moments for Triggering Notifications. *IEEE Pervasive Computing*, *13*(1), 22–29. http://doi.org/10.1109/MPRV.2014.15

Poppinga, B., Heuten, W., & Boll, S. (2014). Sensor-Based identification of opportune Moments for triggering notifications. *Pervasive Computing, IEEE*, *13*(1), 22–29.

Poppinga, B., Oehmcke, S., Heuten, W., & Boll, S. (2013). Storyteller: In-situ Reflection on Study Experiences. In *Proceedings of the 15th International*

Conference on Human-computer Interaction with Mobile Devices and Services (pp. 472–475). New York, NY, USA: ACM. http://doi.org/10.1145/2493190.2494655

Raento, M., Oulasvirta, A., Petit, R., & Toivonen, H. (2005). ContextPhone: A Prototyping Platform for Context-Aware Mobile Applications. *IEEE Pervasive Computing*, *4*, 51–59. http://doi.org/10.1109/MPRV.2005.29

Rahmati, A., & Zhong, L. (2013). Studying Smartphone Usage: Lessons from a Four-Month Field Study. Retrieved from http://ieeexplore.ieee.org.proxy.lib.umich.edu/xpls/abs_all.jsp?arnumber=6212504

Ramanathan, N., Alquaddoomi, F., Falaki, H., George, D., Hsieh, C., Jenkins, J., … Estrin, D. (2012). ohmage: An open mobile system for activity and experience sampling. In *2012 6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)* (pp. 203–204).

Ra, M.-R., Liu, B., La Porta, T. F., & Govindan, R. (2012). Medusa: A programming framework for crowd-sensing applications. In *Proceedings of the 10th international conference on Mobile systems, applications, and services* (pp. 337–350). Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=2307668

Rana, R. K., Chou, C. T., Kanhere, S. S., Bulusu, N., & Hu, W. (2010). Ear-phone: an end-to-end participatory urban noise mapping system. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks* (pp. 105–116). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1791226

Ravi, N., Dandekar, N., Mysore, P., & Littman, M. L. (2005a). Activity recognition from accelerometer data. In *Proceedings of the National*

Conference on Artificial Intelligence (Vol. 20, p. 1541). Retrieved from http://www.aaai.org/Papers/AAAI/2005/IAAI05-013.pdf

Ravi, N., Dandekar, N., Mysore, P., & Littman, M. L. (2005b). Activity recognition from accelerometer data. In *Proceedings of the National Conference on Artificial Intelligence* (Vol. 20, p. 1541). Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Recognizing the User's Current Activity. (n.d.). [http://developer.android.com/training/location/activity-recognition.html].

Reddy, S., Burke, J., Estrin, D., Hansen, M., & Srivastava, M. (2007). A framework for data quality and feedback in participatory sensing. In *Proceedings of the 5th international conference on Embedded networked sensor systems* (pp. 417–418). Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1322324

Reddy, S., Estrin, D., & Srivastava, M. (2010). Recruitment framework for participatory sensing data collections. In *Pervasive Computing* (pp. 138–155). Springer. Retrieved from http://link.springer.com.proxy.lib.umich.edu/chapter/10.1007/978-3-642-12654-3_9

Reddy, S., Mun, M., Burke, J., Estrin, D., Hansen, M., & Srivastava, M. (2010). Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks (TOSN)*, *6*(2), 13.

Reddy, S., Samanta, V., Burke, J., Estrin, D., Hansen, M., & Srivastava, M. (2009). MobiSense—mobile network services for coordinated Participatory Sensing. In *Autonomous Decentralized Systems, 2009. ISADS'09. International Symposium on* (pp. 1–6). Retrieved from http://ieeexplore.ieee.org.proxy.lib.umich.edu/xpls/abs_all.jsp?arnumber=5207328

Reddy, S., Shilton, K., Burke, J., Estrin, D., Hansen, M., & Srivastava, M. (2008). Evaluating participation and performance in participatory sensing. In *International Workshop on Urban, Community, and Social Applications of Networked Sensing Systems. Raleigh, North Carolina, USA* (pp. 1–5). Retrieved from http://www.researchgate.net/publication/229438998_Multimodal_sensing_for_pediatric_obesity_applications/file/d912f50ca604396b1b.pdf#page=7

Reddy, S., Shilton, K., Burke, J., Estrin, D., Hansen, M., & Srivastava, M. (2009). Using context annotated mobility profiles to recruit data collectors in participatory sensing. In *Location and Context Awareness* (pp. 52–69). Springer. Retrieved from http://link.springer.com.proxy.lib.umich.edu/chapter/10.1007/978-3-642-01721-6_4

Reis, H. T., & Gable, S. L. (2000). Event-sampling and other methods for studying everyday experience. *Handbook of Research Methods in Social and Personality Psychology*, 190–222.

Reis, H. T., & Wheeler, L. (1991). Studying social interaction with the Rochester Interaction Record. *Advances in Experimental Social Psychology*, *24*, 269–318.

Ren, J., Zhang, Y., Zhang, K., & Shen, X. S. (2015). SACRM: Social Aware Crowdsourcing with Reputation Management in Mobile Sensing. *Computer Communications*, *65*, 55–65.

Restuccia, F., Das, S. K., & Payton, J. (2015). Incentive Mechanisms for Participatory Sensing: Survey and Research Challenges. *arXiv Preprint arXiv:1502.07687*. Retrieved from http://arxiv.org/abs/1502.07687

Robson, C. (2012). Using Mobile Technology and Social Networking to Crowdsource Citizen Science. Retrieved from http://escholarship.org/uc/item/7pb628kb.pdf

Rogers, Y., Sharp, H., & Preece, J. (2011). *Interaction Design: Beyond Human - Computer Interaction* (3rd ed.). Wiley.

Rosenthal, S., Dey, A. K., & Veloso, M. (2011). Using Decision-Theoretic Experience Sampling to Build Personalized Mobile Phone Interruption Models. In K. Lyons, J. Hightower, & E. M. Huang (Eds.), *Pervasive Computing* (pp. 170–187). Springer Berlin Heidelberg. Retrieved from http://link.springer.com.proxy.lib.umich.edu/chapter/10.1007/978-3-642-21726-5_11

Roy, N., Misra, A., Julien, C., Das, S. K., & Biswas, J. (2011). An energy-efficient quality adaptive framework for multi-modal sensor context recognition. In *Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on* (pp. 63–73). IEEE. Retrieved from http://ieeexplore.ieee.org.proxy.lib.umich.edu/xpls/abs_all.jsp?arnumber=5767596

Sahami Shirazi, A., Henze, N., Dingler, T., Pielot, M., Weber, D., & Schmidt, A. (2014a). Large-scale Assessment of Mobile Notifications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 3055–3064). New York, NY, USA: ACM. http://doi.org/10.1145/2556288.2557189

Sahami Shirazi, A., Henze, N., Dingler, T., Pielot, M., Weber, D., & Schmidt, A. (2014b). Large-scale assessment of mobile notifications. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems* (pp. 3055–3064). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=2557189

Sakamura, M., Yonezawa, T., Nakazawa, J., Takashio, K., & Tokuda, H. (2014). Help Me!: Valuing and Visualizing Participatory Sensing Tasks with Physical Sensors. In *Proceedings of the 2014 International Workshop on Web Intelligence and Smart Sensing* (pp. 3:1–3:6). New York, NY, USA: ACM. http://doi.org/10.1145/2637064.2637095

Salber, D., Dey, A. K., & Abowd, G. D. (1999). The context toolkit: aiding the development of context-enabled applications. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (pp. 434–441). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=303126

Salvador, T., Bell, G., & Anderson, K. (1999). Design Ethnography. *Design Management Journal (Former Series)*, *10*(4), 35–41. http://doi.org/10.1111/j.1948-7169.1999.tb00274.x

Sarker, H., Sharmin, M., Ali, A. A., Rahman, M. M., Bari, R., Hossain, S. M., & Kumar, S. (2014). Assessing the Availability of Users to Engage in Just-in-time Intervention in the Natural Environment. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (pp. 909–920). New York, NY, USA: ACM. http://doi.org/10.1145/2632048.2636082

Sarter, N. (2013). Multimodal support for interruption management: Models, empirical findings, and design recommendations. *Proceedings of the IEEE*, *101*(9), 2105–2112.

Schmidt, A., Takaluoma, A., & Mäntyjärvi, J. (2000). Context-aware telephony over WAP. *Personal Technologies*, *4*(4), 225–229.

Seo, J., Lee, S., & Lee, G. (2011). An experience sampling system for context-aware mobile application development. In *Design, User Experience, and Usability. Theory, Methods, Tools and Practice* (pp. 648–657). Springer. Retrieved from

http://link.springer.com.proxy.lib.umich.edu/chapter/10.1007/978-3-642-21675-6_74

Sheppard, S. A., Wiggins, A., & Terveen, L. (2014). Capturing quality: retaining provenance for curated volunteer monitoring data. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing* (pp. 1234–1245). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=2531689

Shi, J., Zhang, R., Liu, Y., & Zhang, Y. (2010). Prisense: privacy-preserving data aggregation in people-centric urban sensing systems. In *INFOCOM, 2010 Proceedings IEEE* (pp. 1–9). IEEE. Retrieved from http://ieeexplore.ieee.org.proxy.lib.umich.edu/xpls/abs_all.jsp?arnumber=5462147

Shilton, K. (2009). Four Billion Little Brothers?: Privacy, Mobile Phones, and Ubiquitous Data Collection. *Commun. ACM*, *52*(11), 48–53. http://doi.org/10.1145/1592761.1592778

Shilton, K., Burke, J. A., Estrin, D., Hansen, M., & Srivastava, M. (2008). Participatory privacy in urban sensing. *Center for Embedded Network Sensing*. Retrieved from http://escholarship.org/uc/item/90j149pp.pdf

Shin, C., Hong, J.-H., & Dey, A. K. (2012). Understanding and prediction of mobile application usage for smart phones. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (pp. 173–182). New York, NY, USA: ACM. http://doi.org/10.1145/2370216.2370243

Shin, M., Cornelius, C., Peebles, D., Kapadia, A., Kotz, D., & Triandopoulos, N. (2011). AnonySense: A system for anonymous opportunistic sensing. *Pervasive and Mobile Computing*, *7*(1), 16–30. http://doi.org/10.1016/j.pmcj.2010.04.001

Silvertown, J. (2009). A new dawn for citizen science. *Trends in Ecology & Evolution*, *24*(9), 467–471.

Singla, A., & Krause, A. (2013). Incentives for privacy tradeoff in community sensing. In *First AAAI Conference on Human Computation and Crowdsourcing*. Retrieved from http://www.aaai.org/ocs/index.php/HCOMP/HCOMP13/paper/view/7416

Smith, J., & Dulay, N. (2014). RingLearn: Long-term mitigation of disruptive smartphone interruptions. In *2014 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)* (pp. 27–35). http://doi.org/10.1109/PerComW.2014.6815160

Sonnenberg, B., Riediger, M., Wrzus, C., & Wagner, G. G. (2012). Measuring time use in surveys – Concordance of survey and experience sampling measures. *Social Science Research*, *41*(5), 1037–1052. http://doi.org/10.1016/j.ssresearch.2012.03.013

Stevens, M., & D'Hondt, E. (2010). Crowdsourcing of Pollution Data using Smartphones. In *1st Ubiquitous Crowdsourcing Workshop at UbiComp*.

Stikic, M., Van Laerhoven, K., & Schiele, B. (2008). Exploring semi-supervised and active learning for activity recognition. In *Wearable computers, 2008. ISWC 2008. 12th IEEE international symposium on* (pp. 81–88). IEEE. Retrieved from http://ieeexplore.ieee.org.proxy.lib.umich.edu/xpls/abs_all.jsp?arnumber=4911590

Stone, A. A., Bachrach, C. A., Jobe, J. B., Kurtzman, H. S., & Cain, V. S. (1999). *The Science of Self-report: Implications for Research and Practice*. Psychology Press.

Stone, A. A., Shiffman, S., Schwartz, J. E., Broderick, J. E., & Hufford, M. R. (2003). Patient compliance with paper and electronic diaries. *Controlled Clinical Trials*, *24*(2), 182–199.

Sun, Y., Zhu, Y., Feng, Z., & Yu, J. (2014). Sensing processes participation game of smartphones in participatory sensing systems. In *Sensing, Communication, and Networking (SECON), 2014 Eleventh Annual IEEE International Conference on* (pp. 239–247). IEEE. Retrieved from http://ieeexplore.ieee.org.proxy.lib.umich.edu/xpls/abs_all.jsp?arnumber=6990359

Ter Hofte, G. H. (2007). What's that hot thing in my pocket? SocioXensor, a smartphone data collector. *Proceedings of the E-Social Science*, 7–9.

Thebault-Spieker, J. (2012). Crowdsourced Participatory Sensing: applications and motivation of work.

Thiagarajan, A., Biagioni, J., Gerlich, T., & Eriksson, J. (2010). Cooperative transit tracking using smart-phones. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems* (pp. 85–98). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1869993

Thiagarajan, A., Ravindranath, L., LaCurts, K., Madden, S., Balakrishnan, H., Toledo, S., & Eriksson, J. (2009). VTrack: accurate, energy-aware road traffic delay estimation using mobile phones. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems* (pp. 85–98). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1644048

Tomasic, A., Zimmerman, J., Steinfeld, A., & Huang, Y. (2014). Motivating Contribution in a Participatory Sensing System via Quid-pro-quo. In *Proceedings of the 17th ACM Conference on Computer Supported*

Cooperative Work & Social Computing (pp. 979–988). New York, NY, USA: ACM. http://doi.org/10.1145/2531602.2531705

Truskinger, A., Yang, H., Wimmer, J., Zhang, J., Williamson, I., & Roe, P. (2011). Large Scale Participatory Acoustic Sensor Data Analysis: Tools and Reputation Models to Enhance Effectiveness. In *2011 IEEE 7th International Conference on E-Science (e-Science)* (pp. 150–157). http://doi.org/10.1109/eScience.2011.29

Turner, L. D., Allen, S. M., & Whitaker, R. M. (2015). Interruptibility Prediction for Ubiquitous Systems: Conventions and New Directions from a Growing Field. Retrieved from https://users.cs.cf.ac.uk/L.Turner/publication_files/ubicomp15_interruptibility.pdf

Vice-Chair, D. of P. & B. S. S. B. U. A. S. P. and, Pittsburgh, P. S. S. R. P. of C. & H. P. of, Institute, D. of C. C. & P. S. A. A. P. D. Behavioral Research Program Health Promotion Branch National Cancer, & Institute, P. R. B. L. N. C. Behavioral Research Program National Cancer. (2007). *The Science of Real-Time Data Capture : Self-Reports in Health Research: Self-Reports in Health Research*. Oxford University Press.

Villanueva, F. J., Villa, D., Santofimia, M. J., Barba, J., & Lopez, J. C. (2015). Crowdsensing smart city parking monitoring. In *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on* (pp. 751–756). IEEE. Retrieved from http://ieeexplore.ieee.org.proxy.lib.umich.edu/xpls/abs_all.jsp?arnumber=7389148

Vondrick, C., Patterson, D., & Ramanan, D. (2012). Efficiently Scaling up Crowdsourced Video Annotation. *International Journal of Computer Vision*, *101*(1), 184–204. http://doi.org/10.1007/s11263-012-0564-1

Wang, X., Cheng, W., Mohapatra, P., & Abdelzaher, T. (2013). Artsense: Anonymous reputation and trust in participatory sensing. In *INFOCOM, 2013 Proceedings IEEE* (pp. 2517–2525). IEEE. Retrieved from http://ieeexplore.ieee.org.proxy.lib.umich.edu/xpls/abs_all.jsp?arnumber=6567058

Wang, Y., Chen, R., & Wang, D.-C. (2015). A survey of mobile cloud computing applications: Perspectives and challenges. *Wireless Personal Communications*, *80*(4), 1607–1623.

Weiser, M. (1991). The computer for the 21st century. *Scientific American*, *265*(3), 94–104.

Welbourne, E., Balazinska, M., Borriello, G., & Fogarty, J. (2010). Specification and Verification of Complex Location Events with Panoramic. *Pervasive Computing*, *6030*, 57–75.

Winograd, T. (2001). Architectures for context. *Human-Computer Interaction*, *16*(2), 401–419.

Xiang, C., Li, X., Yang, P., Tian, C., & Li, Q. (2013). Feeling Sensors' Pulse: Accurate Noise Quantification in Participatory Sensing Network. In *Mobile Ad-hoc and Sensor Networks (MSN), 2013 IEEE Ninth International Conference on* (pp. 212–219). IEEE. Retrieved from http://ieeexplore.ieee.org.proxy.lib.umich.edu/xpls/abs_all.jsp?arnumber=6726333

Xiang, C., Yang, P., Tian, C., Cai, H., & Liu, Y. (2015). Calibrate without Calibrating: An Iterative Approach in Participatory Sensing Network. *IEEE Transactions on Parallel and Distributed Systems*, *26*(2), 351–361. http://doi.org/10.1109/TPDS.2014.2308205

Xiao, Y., Simoens, P., Pillai, P., Ha, K., & Satyanarayanan, M. (2013). Lowering the Barriers to Large-scale Mobile Crowdsensing. In *Proceedings of the*

14th Workshop on Mobile Computing Systems and Applications (pp. 9:1–9:6). New York, NY, USA: ACM. http://doi.org/10.1145/2444776.2444789

Xu, J. Y., Pottie, G. J., & Kaiser, W. J. (2013). Enabling Large-Scale Ground-Truth Acquisition and System Evaluation in Wireless Health. *Biomedical Engineering, IEEE Transactions on*, *60*(1), 174–178.

Xu, Q., Erman, J., Gerber, A., Mao, Z., Pang, J., & Venkataraman, S. (2011). Identifying Diverse Usage Behaviors of Smartphone Apps. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference* (pp. 329–344). New York, NY, USA: ACM. http://doi.org/10.1145/2068816.2068847

Yang, H., Zhang, J., & Roe, P. (2011). Using reputation management in participatory sensing for data classification. *Procedia Computer Science*, *5*, 190–197.

Yoon, S., Lee, S., Lee, J., & Lee, K. (2014). Understanding Notification Stress of Smartphone Messenger App. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems* (pp. 1735–1740). New York, NY, USA: ACM. http://doi.org/10.1145/2559206.2581167

Zhang, D., Xiong, H., Wang, L., & Chen, G. (2014). CrowdRecruiter: Selecting Participants for Piggyback Crowdsensing Under Probabilistic Coverage Constraint. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (pp. 703–714). New York, NY, USA: ACM. http://doi.org/10.1145/2632048.2632059

Zhang, X., Gong, H., Xu, Z., Tang, J., & Liu, B. (2012). Jam eyes: a traffic jam awareness and observation system using mobile phones. *International Journal of Distributed Sensor Networks*, *2012*. Retrieved from http://www.hindawi.com.proxy.lib.umich.edu/journals/ijdsn/2012/921208/abs/

Zhou, P., Zheng, Y., & Li, M. (2012). How Long to Wait?: Predicting Bus Arrival Time with Mobile Phone Based Participatory Sensing. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services* (pp. 379–392). New York, NY, USA: ACM. http://doi.org/10.1145/2307636.2307671

Zhuang, Z., Kim, K.-H., & Singh, J. P. (2010). Improving energy efficiency of location sensing on smartphones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services* (pp. 315–330). Retrieved from http://www.deutsche-telekom-laboratories.com/~kyuhan/papers/MobiSys10Kim.pdf

Zhu, Y., Li, Z., Zhu, H., Li, M., & Zhang, Q. (2013). A compressive sensing approach to urban traffic estimation with probe vehicles. *Mobile Computing, IEEE Transactions on*, *12*(11), 2289–2302.

Zimmerman, J., Tomasic, A., Garrod, C., Yoo, D., Hiruncharoenvate, C., Aziz, R., … Steinfeld, A. (2011a). Field trial of tiramisu: crowd-sourcing bus arrival times to spur co-design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1677–1686). ACM. Retrieved from http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1979187

Zimmerman, J., Tomasic, A., Garrod, C., Yoo, D., Hiruncharoenvate, C., Aziz, R., … Steinfeld, A. (2011b). Field trial of Tiramisu: crowd-sourcing bus arrival times to spur co-design. In *Proceedings of the 2011 annual conference on Human factors in computing systems* (pp. 1677–1686). New York, NY, USA: ACM. http://doi.org/10.1145/1978942.1979187